1986-12

Modeling and experimental validation of a single-link flexible manipulator.

Gannon, Kevin P.

http://hdl.handle.net/10945/21981

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

MODELING AND EXPERIMENTAL VALIDATION OF A SINGLE-LINK FLEXIBLE MANIPULATOR

by

Kevin P. Gannon

December 1986

Thesis Advisor: L. W. Chang

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL *(If applicable)* 69	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS *(City, State, and ZIP Code)* Monterey, California 93943-5000		7b ADDRESS *(City, State, and ZIP Code)* Monterey, California 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)*

MODELING AND EXPERIMENTAL VALIDATION OF A SINGLE-LINK FLEXIBLE MANIPULATOR

12 PERSONAL AUTHOR(S)

GANNON, Kevin P.

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM ________ TO ______	14 DATE OF REPORT *(Year, Month Day)* 1986 December	15 PAGE COUNT 97

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

Flexible Manipulator, Robotics

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Structural flexibility effects on robot mechanisms is a very important consideration in the move toward lighter, faster, and more accurate robot systems. This thesis presents the computer simulation and experimental validation of a dynamic model of an experimental single-link flexible manipulator, using the Equivalent Rigid Link System with an enhanced natural-mode discretization. The experimental arm is driven by a hydraulic actuator and moves in a vertical plane. Hydraulic actuator dynamics and the effects of gravity are included in the equations of motion. Computer simulation for the experimental arm is performed by the Dynamic Simulation Language (DSL). The validation of the dynamic model includes the comparison between the actual tip position and the predicted arm tip position.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Chang, Code 69Ck	22b TELEPHONE *(Include Area Code)* 2632	22c OFFICE SYMBOL 69Ck

83 APR edition may be used until exhausted
All other editions are obsolete

Modeling And Experimental Validation of a Single-Link Flexible Manipulator

by

Kevin P. Gannon
Lieutenant, United States Navy
B.S., Tulane University, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

ABSTRACT

Structural flexibility effects on robot mechanisms is a very important consideration in the move toward lighter, faster, and more accurate robot systems. This thesis presents the computer simulation and experimental validation of a dynamic model of an experimental single-link flexible manipulator, using the Equivalent Rigid Link System with an enhanced natural-mode discretization. The experimental arm is driven by a hydraulic actuator and moves in a vertical plane. Hydraulic actuator dynamics and the effects of gravity are included in the equations of motion. Computer simulation for the experimental arm is performed by the Dynamic Simulation Language (DSL). The validation of the dynamic model includes the comparison between the actual tip position and the predicted arm tip position.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered completely validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I wish to extend my sincerest thanks and appreciation to my advisor, Professor Liang-Wey Chang for his steadfast support and guidance during this trying and rewarding period. I would also like to thank Professor Wu of the Aeronautical Engineering Department of the Naval Postgraduate School for his generous technical equipment support.

I. INTRODUCTION

A. MOTIVATION

As the Navy approaches the 21st century, the potential for robotic technology is increasing. Present and possible future applications range from industrial welding to shipboard firefighting, from undersea exploration and salvage to weapons handling. [Ref. 1: pp. 11-14] All offer the promise of freeing the sailor from boring, repetitive tasks or, more importantly, making hazardous jobs much safer. Although Naval applications will most likely involve some redesign or adaptation from their civilian counterpart in order to meet the demands of an at-sea environment, it is necesary to keep abreast of the technological advancements as they occur. Everett [Ref. 1] presents and excellent overview of current development efforts and in the process shows robotics to be an inevitable part of the Navy's future.

B. NATURE OF THE PROBLEM

Earlier work in robotics concentrated strictly on a rigid body model of a manipulator system. Determination of the equations of motion, inverse dynamics, and end-effector control are examples of the types of problems that have been solved for the rigid body case [Ref. 2: p.11]. However, this rigid body assumption led to manipulators that are characterized by their large size, both in cross-sectional area and weight, slow operational speed and low payload to total weight ratio.

The demand for higher speeds and better system performance have caused the investigation of lightweight designs, most notably in space applications [Ref. 3: pp.3,9]. The advantages of the lightweight design include increase the speed of operation, and the ease of transportability, decreased energy consumption, and lowered overall cost [Ref. 4: p.167].

However, all of these advantages are not without cost. The lightweight design requires that the flexibility effects should not be neglected. The degradation of end-effector positioning accuracy occurs if the increased deformations due to bending and torsion effects of the lightweight, flexible manipulator are not modeled. (Flexibility must be taken into account since actuators which actuate rigid motion also excite deformations.)

Additional information is needed regarding the dynamics of the flexible manipulator so that the system can be designed to withstand loads without exceeding fatigue and bending stress limits. In addition, system natural frequencies are needed for control and sampling considerations [Ref. 5: p. 2]. A method for dynamic modeling of manipulators with flexible members is needed for the design of controllers. This model must be able to accurately predict system responses, be able to be efficiently simulated and be computationally effective so that accurate motion control can be achieved.

C. LITERATURE REVIEW

Research into the design and control of flexible manipulators began in the early to mid 1970's. The efforts centered around the design, modeling and control of a planar two link manipulator. Since then, many approaches have been taken to the development of an accurate dynamic model.

Most models involve an attempt to find the relationship between the large, rigid-body motion (large motion) and the small motion displacements (small motion) arising from the flexibility of the structure.

Book [Ref. 6] used the 4 × 4 transformation matrices, common to rigid body models, to model the elastic chain. This enabled the representation of spatial motion of a manipulator with an arbitrary number of links and joints. His model was limited in the assumption that the mass of the manipulator is negligible compared to the mass of the load. This is acceptable in space applications where the manipulator is very light and flexible, but not for industrial applications where the effects of mass must be considered.

Sunada and Dubowsky [Ref. 7] similiarly expressed the kinematics and dynamics of the manipulator in terms of 4 × 4 transformation matrices. Lagrangian formulation with finite element methods was used to model the system with flexibility effects. A dynamic coordinate reduction procedure, Component Mode Synthesis, was used to reduce the number of governing equations improving computational efficiency. Manipulator motion was expressed as "small perturbations about some known nominal motion of the system". This small motion superposition onto an assumed nominal large motion neglects the effects of the small motion interaction on the large motion; in turn, the consideration of the coupling effect was not complete.

Book [Refs. 8,9] also includes small motion displacements in the 4 × 4 transformation matrices. He used a modal approach to model system flexible kinematics, truncating the series of assumed vibration modes. After applying recursive Lagrangian dynamics and a hybrid set of large and small motion coordinates, the dynamic equations of motion were obtained. Although an accurate model describing large motion, small motion and their coupling, the resulting set of coupled, non-linear second order ordinary differential equations is computationally intensive and, therefore, expensive to solve.

Truckenbrodt [Ref. 10] developed a model for a six degree of freedom manipulator with two flexible links. The flexible manipulator was modeled as a "hybrid multibody system" having both rigid and flexible elements. The flexible coordinates were represented using truncated series expansion. The model was then linearized for control purposes.

Cannon and Schmitz [Ref. 11] performed experimental studies on a "very flexible" single link manipulator. Their model was also truncated and linearized. A simplified analytical model was used to derive the general form of the equations of motion. Experimental analysis was then used to determine the coefficients.

Chang [Ref. 5] introduced the Equivalent Rigid Link System (ERLS) dynamic model of a flexible manipulor. Global motions were separated into large motions and small motions. The ERLS described the kinematics of the large motion. The small motion displacements were described relative to the ERLS. The finite element method was used to discretize deformations and Lagrangian formulation was used to derive the equations of motion. Two sets of coupled, non-linear ordinary differential equations-large motion equations and small motion equations resulted. The set of large motion equations were non-linear in both the large and small motion variables. The set of small motion equations were linear in the small motion variable but non-linear in the large motion variable. Chang also developed a solution technique called the Sequential Integration Method. It allows simulations of systems with inertia coupled motions having non-linear slow motion (large motion) with linear fast motion (small motion). Chang's model presented a complete, efficient dynamic model being able to describe large motion, small motion and their coupling.

Petroka [Ref. 12] performed an experimental validation of Chang's dynamic model. He designed a hydraulically actuated, single-link flexible arm and applied the ERLS model. He tailored the model for a single link flexible manipulator with vertical plane motion. Finite element analysis was limited by single element modeling and the choice of a simple cubic function representation for the shape function. Tip position information was determined from motion picture studies, additional data was obtained from a comparison between theoretical and experimental strain. Results of the validation suggested the potential usefulness of the model in determining tip position.

D. PURPOSE

The purpose of this study is to perform a computer simulation and experimental validation of a dynamic model including flexibility. Chang's Equivalent Rigid Link System dynamic model using an enhanced natural-mode discretization is chosen. The research also serves as a continuation of the experimental work performed on the single-link flexible manipulator designed by Petroka. Computer simulation for the experimental arm is performed by the Dynamic Simulation Language (DSL) on the IBM 3033 mainframe computer. A piezo-resistive accelerometer and strain gauges are used for tip position and strain information. Data acquisition was performed using a GWI Instruments MacAdios hardware and software data acquisition system in conjunction with a Macintosh 512k computer.

An understanding of the Equivalent Rigid Link System model is essential. The theory and application behind the model, its adaptation to the single link case, the natural-mode discretization, and a discussion of the computer simulation, are discussed in Chapter II.

A description of the experimental apparatus, design and procedure are reviewed in Chapter III. Data acquisition is also discussed.

Chapter IV presents the results of the validation. Comparisons are made between experimental arm-tip position, theoretical arm-tip position using cubic shape functions, and theoretical arm-tip position using natural-mode discretization. Also a comparison is made between experimental strain and the theoretical strain predicted by the ERLS dynamic model simulation.

Chapter V is devoted to conclusions reached during the study and to recommendations for future work in the area of flexible manipulators.

II. MODEL FORMULATION FOR A SINGLE-LINK FLEXIBLE MANIPULATOR

A. EQUIVALENT RIGID LINK SYSTEM

The ERLS dynamic model of a flexible manipulator describes the large rigid-body motion of the manipulator and the small-motion displacement arising from the flexibility of the structure. A schematic of the ERLS is shown in Fig. 2.1. The ERLS describes the large motion of the manipulator. A strength of the ERLS is that the description of kinematics is equivalent to the rigid manipulator system so that the the familiar mathematical tools for the rigid system can be used to solve flexible manipulator problems. The small-motion displacements are described relative to the ERLS. Displacements are for each point along the flexible arm,a function of location and time, so it is necessary to discretize the deformations for digital computation. The techniques of the Finite Element Method (FEM) are utilized for this discretization. In this study, an enhanced natural-mode discretization which represents the small motion in terms of nodal displacements and the natural modes of the structure is introduced.

After describing the kinematic relationships between the large and small motions, kinetics is introduced to complete the derivation of the equations of motion. The Lagrangian dynamics approach is used for the derivation due to its straightforward and

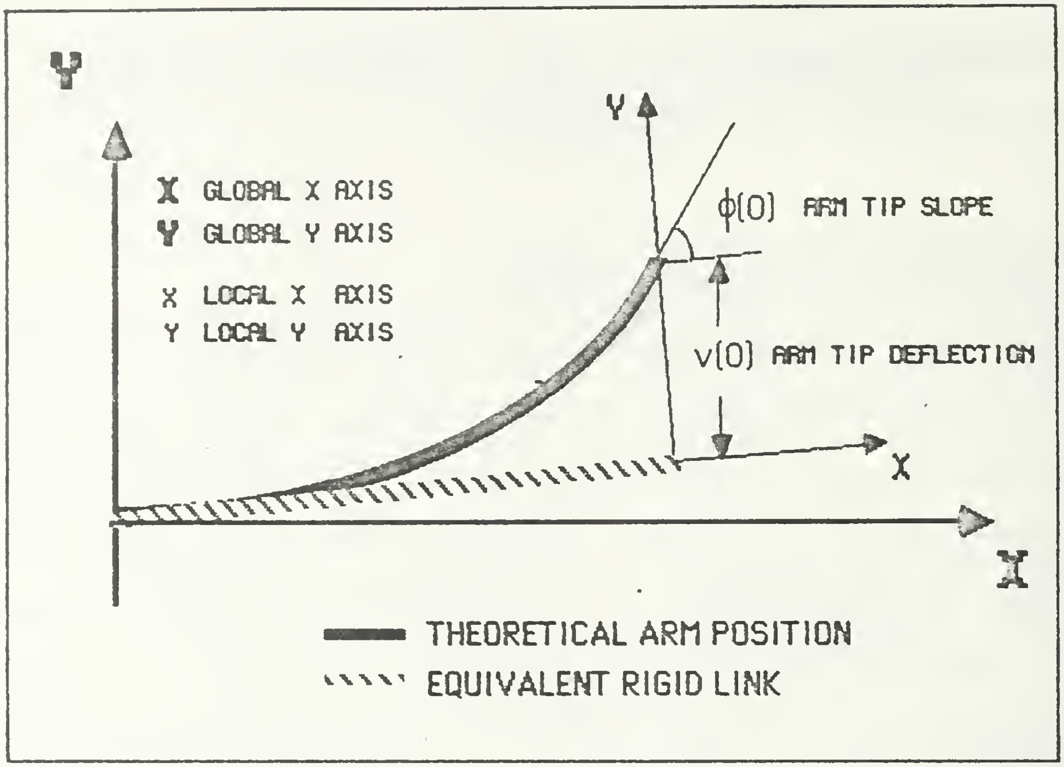

Figure 2.1 Equivalent Rigid Link System (ERLS)

systematic nature which is helpful in the analysis of complex systems. This formulation requires generalized coordinates. Since the global motions are separated into large motions and small motions, generalized coordinates are defined to represent these two motions. The three generalized coordinates chosen are the rigid body rotation, θ, and two nodal displacements, v(0) and ϕ(0), which can be measured at the end of the link. Kinetic energy of the system is due to kinetic energy of the link, kinetic energy of the actuator and the kinetic energy of any load. The actuator is treated as a rigid body and all kinetic energy terms are calculated separately. Potential energy of the system comes from strain energy due to deformation and from gravity. Generalized forces include any applied forces and damping forces. The actual derivation of the equations of motion requires detailed, tedious mathematical manipulations, and the Lagrangian formulation yields two sets of equations. One set describes the large motions and the other set describes the small motions. These two sets of equations are non-linear, coupled, second-order, ordinary differential equations represented as follows,

$$M_{qq} \ddot{\theta} + M_{q\eta} \ddot{U} - F_q \qquad (2\text{-}1)$$

$$M_{\eta q} \ddot{\theta} + M_{\eta\eta} \ddot{U} + K_\eta U = F_\eta \qquad (2\text{-}2)$$

where

M_{qq} - 1×1 effective inertia matrix for large motions

$M_{q\eta}$ - 1×2 coupled inertia matrix of the small motion effect on large motion

$M_{\eta q}$ - 2×1 coupled inertia matrix of the large motion effect on small motions

$M_{\eta\eta}$ - 2×2 effective inertia matrix for small motions

K_η - 2×2 stiffness matrix for small motions

F_q - 1×1 load vector for the large motions

F_η - 2×1 load vector for the small motion

$\ddot{\theta}$ - generalized coordinate of the large motions

U - 2×1 vector ,generalized coordinate of the deformations representing the small motions

A more detailed derivation of the equations of motion is included in Appendix A.

B. SHAPE FUNCTION DERIVATION

In Petroka's study a cubic shape function was assumed to represent the transverse displacement of the flexible manipulator. This allowed a complete representation of the displacement including rigid body rotation and translation. In this study the natural-mode shape functions of a beam are used to represent the flexural motion of the single-link flexible arm. Only the first two mode shapes are used. The flexible manipulator arm is modeled as a continuous Euler-Bernoulli cantilever beam, neglecting shear deformation, and rotary effects. The beam has the following characteristics,

μ=mass/unit length

EI=flexural rigidity

M=moment

$\partial M/\partial x = V$

L=length

p=transverse load/unit length

V = shear force

$\partial V/\partial x = p$

k=curvature = $M/EI = \partial^2 z/\partial x^2/\{1+(\partial y/\partial x)^2\}^{3/2} \approx \partial^2 z/\partial x^2$.

(linear beam theory)

This allows us to express p in the following form,

$$p = \partial^2 M/\partial x^2 = \partial^2[EI\ \partial^2 \nu/\partial x^2]/\partial x^2 = -\mu\ \partial^2 \nu/\partial t^2 \qquad (2\text{-}3)$$

Thus, for a uniform beam,

$$EI\ \partial^4 \nu/\partial x^4 = -\mu\ \partial^2 \nu/\partial t^2 \qquad (2\text{-}4)$$

Let $\nu(x,t) = X(x) \sin \omega t$, substituting into Eq. (2–4),

$$EI\ X^{IV} \sin \omega t = -\mu\ \omega^2 X \sin \omega t$$

therefore,

$$X^{IV} = \beta^4 X \qquad \text{where} \quad \beta^4 = \mu\ \omega^2/EI \qquad (2\text{-}5)$$

The above equation has the general solution,

$$X = A \cosh \beta x + B \sinh \beta x + C \cos \beta x + D \sin \beta x \qquad (2\text{-}6)$$

For the cantilever beam model the boundary conditions require that,

$$\text{at } x = -L \text{ (base)}, \qquad X\ (-L) = 0 \quad \text{and} \quad X'\ (-L) = 0 \qquad (2\text{-}7)$$

$$\text{at } x = 0 \ \ \text{(tip)} \qquad X''(0) = 0 \quad \text{and} \quad X'''\ (0) = 0 \qquad (2\text{-}8)$$

Rewriting Eq. (2–6) in the form,

$$\begin{aligned} X = {} & A [\cos \beta x + \cosh \beta x] + B [\cos \beta x - \cosh \beta x] \\ & + C [\sin \beta x + \sinh \beta x] + D [\sin \beta x - \sinh \beta x] \qquad (2\text{-}9) \end{aligned}$$

shows that, (2–7) implies B=D=0, consequently,

$$X = A [\cos \beta x + \cosh \beta x] + C [\sin \beta x + \sinh \beta x] \qquad (2\text{-}10)$$

Using the boundary conditions from Eq. (2-7) and substituting into Eq. (2-10) and solving yields:

$$\cosh \beta L \cos \beta L = -1 \qquad (2\text{-}11)$$

The first three consecutive roots are:

$$\beta_1 L = 1.875104069$$

$$\beta_2 L = 4.694091133$$

$$\beta_3 L = 7.854757438$$

Equation (2-10) is now written in the form:

$$X = A_i' \,(\cos \beta x + \cosh \beta x] + (\sin \beta x + \sinh \beta x] \qquad (2\text{-}12)$$

where $A_i' = \dfrac{\sin \beta_i L + \sinh \beta_i L}{\cos \beta_i L + \cosh \beta_i L} = A/C$

This form satisfies Eqs. (2-7), (2-8), and (2-11) and will be the form used in further derivations.

The transverse displacement, and slope, of the single link flexible arm can now be represented in the following forms, respectively,

$$\begin{aligned} V(x,t) &= a_1(t)\, X_1(x) + a_2(t)\, X_2(x) \qquad (2\text{-}13)\\ &= a_1 (A_1'' (\cos \beta_1 x + \cosh \beta_1 x] + (\sin \beta_1 x + \sinh \beta_1 x]) \\ &\quad + a_2 (A_2' (\cos \beta_2 x + \cosh \beta_2 x] + (\sin \beta_2 x + \sinh \beta_2 x]) \end{aligned}$$

$$\begin{aligned} \phi &= a_1 (A_1' \beta_1 (-\sin \beta_1 x + \sinh \beta_1 x] + \beta_1 (\cos \beta_1 x + \cosh \beta_1 x]) \qquad (2\text{-}14)\\ &\quad + a_2 (A_2' (\beta_2 (-\sin \beta_2 x + \sinh \beta_2 x] + \beta_2 (\cos \beta_2 x + \cosh \beta_2 x]) \end{aligned}$$

Substituting the boundary conditions into the shape functions gives,

$$v(0) = 2\,a_1(t)\,A_1' + 2\,a_2(t)\,A_2' \qquad (2\text{–}15)$$

$$\phi(0) = 2\,a_1(t)\,\beta_1 + 2\,a_2(t)\,\beta_2 \qquad (2\text{–}16)$$

$$v(-L) = 0 \qquad (2\text{–}17)$$

$$\phi(-L) = 0 \qquad (2\text{–}18)$$

Solving the two equations for a_1 and a_2 gives:

$$a_1(t) = \frac{2\,\beta_2\,v(0) - 2\,A_2'\,\phi(0)}{4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1} \quad a_2(t) = \frac{-2\,\beta_1\,v(0) + 2\,A_1'\,\phi(0)}{4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1} \qquad (2\text{–}19)$$

For ease of manipulation, make the following substitutions:

$$C_1 = 2\,\beta_2 / (4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1) \qquad (2\text{–}20)$$

$$C_2 = -2\,\beta_1 / (4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1) \qquad (2\text{–}21)$$

$$C_3 = -2\,A_2' / (4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1) \qquad (2\text{–}22)$$

$$C_4 = 2\,A_1' / (4\,A_1'\,\beta_2 - 4\,A_2'\,\beta_1) \qquad (2\text{–}23)$$

Substituting Eqs. (2–20), (2–21), (2–22), and (2–23) into Eq. (2–13) and collecting terms yields an expression for the transverse displacement of the single–link flexible arm as a function of the arm tip nodal displacements, $v(0)$ and $\phi(0)$:

$$\begin{aligned} v(x,t) = [\, & C_1 [\, A_1' (\cos\beta_1 x + \cosh\beta_1 x) + (\sin\beta_1 x + \sinh\beta_1 x)] \qquad (2\text{–}24) \\ & + C_2 [\, A_2' (\cos\beta_2 x + \cosh\beta_2 x) + (\sin\beta_2 x + \sinh\beta_2 x)]]\, v(0) \\ & + [\, C_3 [\, A_1' (\cos\beta_1 x + \cosh\beta_1 x) + (\sin\beta_1 x + \sinh\beta_1 x)] \\ & + C_4 [\, A_2' (\cos\beta_2 x + \cosh\beta_2 x) + (\sin\beta_2 x + \sinh\beta_2 x)]]\, \phi(0) \end{aligned}$$

This expression is differentiated twice to obtain $v''(x)$, which is necessary for the calculation of the potential energy due to deformation and the theoretical strain,

$$\begin{aligned} v''(x,t) = (\, & C_1 \beta_1^2 (A_1' (-\cos \beta_1 x + \cosh \beta_1 x) + (-\sin\beta_1 x + \sinh \beta_1 x)) \\ & + C_2 \beta_2^2 (A_2' (-\cos \beta_2 x + \cosh \beta_2 x) + (-\sin \beta_2 x + \sinh \beta_2 x))) \, v(0) \\ & + (C_3 \beta_1^2 (A_1' (-\cos \beta_1 x + \cosh \beta_1 x) + (-\sin\beta_1 x + \sinh \beta_1 x)) \\ & + C_4 \beta_2^2 (A_2' (-\cos \beta_2 x + \cosh \beta_2 x) + (-\sin \beta_2 x + \sinh \beta_2 x))) \, \phi(0) \end{aligned} \tag{2-25}$$

Now substitution of $v(x)$ into the 3×1 deformation vector, d, yields the 3×2 shape function matrix, Φ , and the 2×1 nodal displacement vector, U:

$$d = \Phi \, U \qquad \Leftrightarrow \qquad \begin{pmatrix} 0 \\ 0 \\ v(x) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ a & b \end{pmatrix} \begin{pmatrix} v(0) \\ \phi(0) \end{pmatrix} \tag{2-26}$$

where a and b are the coefficients of $v(0)$ and $\phi(0)$, respectively, in Eq. (2-24), defined here for convenience,

$$\begin{aligned} a = \; & C_1 (A_1' (\cos \beta_1 x + \cosh \beta_1 x) + (\sin\beta_1 x + \sinh \beta_1 x)) \\ & + C_2 (A_2' (\cos \beta_2 x + \cosh \beta_2 x) + (\sin \beta_2 x + \sinh \beta_2 x)) \end{aligned} \tag{2-27}$$

$$\begin{aligned} b = \; & C_3 (A_1' (\cos \beta_1 x + \cosh \beta_1 x) + (\sin\beta_1 x + \sinh \beta_1 x)) \\ & + C_4 (A_2' (\cos \beta_2 x + \cosh \beta_2 x) + (\sin \beta_2 x + \sinh \beta_2 x)) \end{aligned} \tag{2-28}$$

The shape function matrix is now in a form convenient for computer coding.

C. HYDRAULIC ACTUATION

The single-link flexible manipulator uses hydraulic actuation. Therefore, hydraulic actuator and servovalve dynamics are an integral part of the total system model. Figure 2.2 illustrates the relationship of hydraulic dynamics to the overall system.

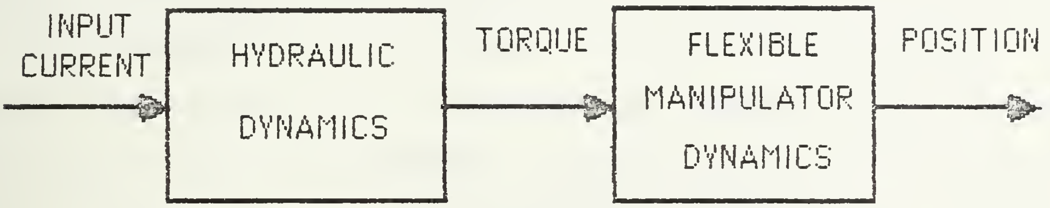

Figure 2.2. System drawing

The hydraulic dynamics are represented by servovalve dynamics and actuator dynamics. Moog, the manfacturer of the servovalve, simplified the description of servovalve dynamics to a single equation [Ref. 13],

$$Q = I K \sqrt{P_v} \tag{2-29}$$

where

Q = Flow delivered from servovalve

I = Input current

K = Valve sizing constant ,contributes to hydraulic system damping

P_v = Valve pressure drop, $P_S - P_L$

Actuator dynamics consist of a form of the continuity eqation (2-30) and the torque output equation (2-31) [Ref. 14],

$$Q = D_m \dot{\theta} + C_{tm} P_L + \frac{V_t P_L}{4 \beta_e} \qquad (2\text{-}30)$$

$$T_d = \eta_t P_L D_m \qquad (2\text{-}31)$$

where

Q	=	Flow delivered to actuator
$D_m \dot{\theta}$	=	Flow causing actuator rotation
$C_{tm} P_L$	=	Leakage flow in actuator
$\frac{V_t \dot{P}}{4 \beta_e}$	=	Compressibility flow
T_d	=	Torque required to overcome inertia and move the load
η_t	=	Torque efficiency
P_L	=	Load pressure drop
D_m	=	Motor displacement

A detailed account into the selection of hydraulic components for the system was included in Petroka's work.

D. COMPUTER SIMULATION

Computer simulation of the equations of motion is performed by the Dynamic Simulation Language (DSL) which solves a set of simultaneous, nonlinear, second-order, coupled, ordinary differential equations. DSL is a differential equation solver that was designed by IBM for mini and mainframe computers as a follow on to Continuous System Modeling Program (CSMP). It was chosen for its compatibility with Fortran, good selection of integration methods (Adams method chosen), implicit double precision, powerful graphics capability, and ease of use.

The computer code was adapted from Petroka's work. Two major differences are Petroka used CSMP and a variable step, fourth order, Runge-Kutta integration method. The decision to use the Adams method was based on several trial runs of the program. In each run the Adams method was significantly faster than Runge-Kutta, sometimes as high as five times faster yet maintaining agreement of solution to the fourth decimal place.

The algorithm of solution involves substituting Eqs. (2-1) and (2-2) into matrix form to create a 3 × 3 coefficient inertia matrix and a 3 × 1 right hand side vector of forces and moments. The unknowns are $\ddot{\theta}$ and $\ddot{U}$.

Each coefficient is calculated in a separate subroutine. Once calculated the elements are put into matrix form. An IMSL linear equation solver subroutine is then called to solve for the accelerations. The accelerations are then integrated twice using the Adams method of the DSL system. A global transformation subroutine is then called to global arm tip position. The computer code is included as Appendix B.

III. EXPERIMENTAL FLEXIBLE ARM

A. EXPERIMENTAL APPARATUS

The experimental apparatus used to validate the Equivalent Rigid Link System (ERLS) on a single-link flexible manipulator consisted of a flexible arm, a hydraulic system, sensors, an analog to digital converter, and a computer.

1. Flexible Arm

The flexible arm shown in Figures 3.1. and 3.2. is a one meter long flexible structure which can bend freely in the vertical plane but is stiff in torsion and horizontal bending. The arm consists of two parallel steel flat bars welded at the base and directly clamped to the hydraulic actuator. Torsional stiffness is obtained by connecting the two steel beams, with thin steel strips, to seven transverse steel bridges. If the arm consisted of only a single steel beam it would deflect due to any bending moment and twist around the beam neutral axis. The additional thin steel strips act to increase the torsional rigidity of the beam.

Table 3.1 shows some geometric and mass properties of the flexible arm.

2. Hydraulic System

The power for the flexible arm was provided by a hydraulic system . The system consisted of a York hydraulic

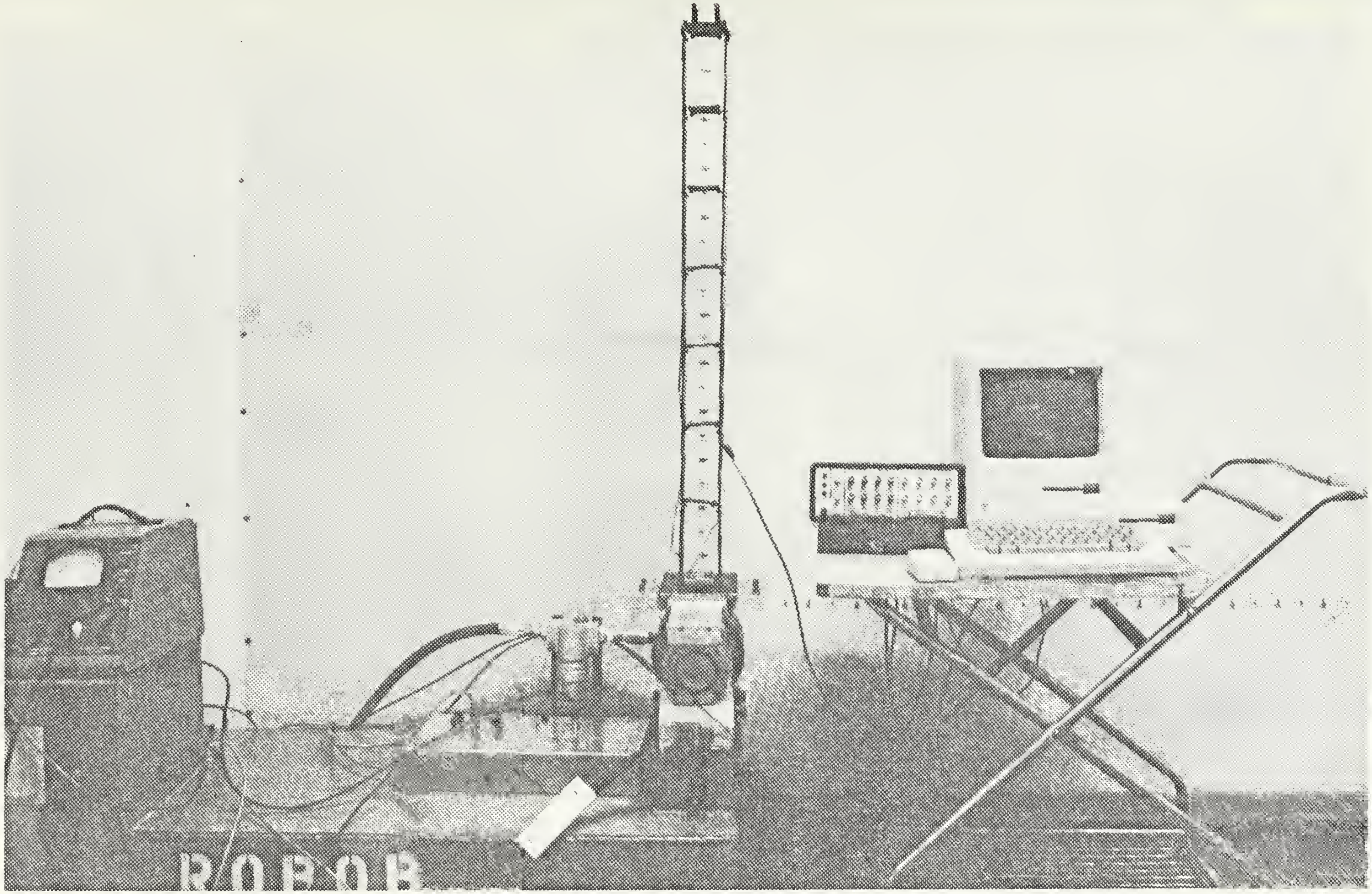

Figure 3.1. Single-Link Flexible Manipulator System

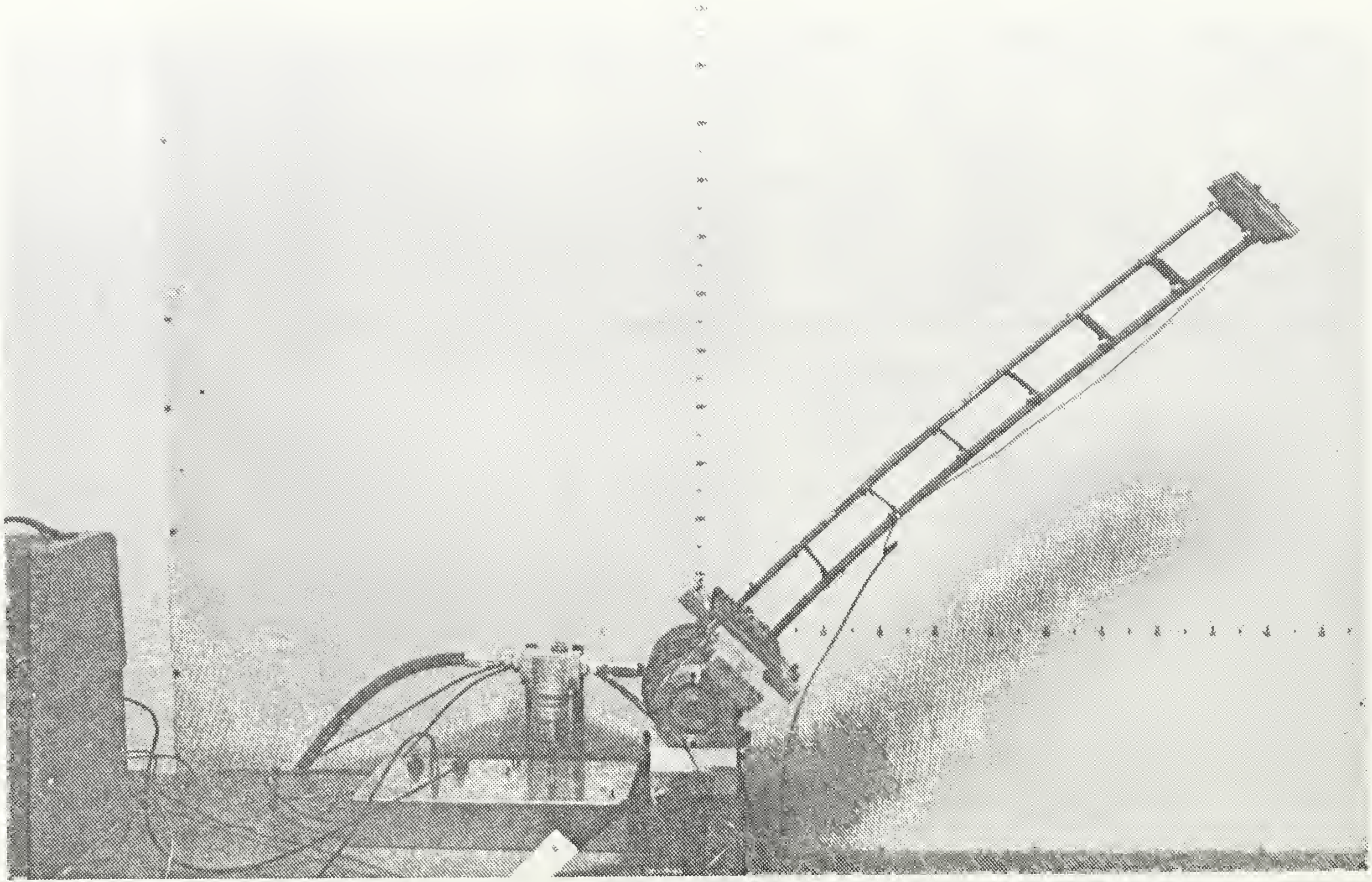

Figure 3.2. Manipulator in Intermediate Position (2.115 Kg Load)

TABLE 3.1

ARM PARAMETERS

Parameter	Value
LL=arm length	0.9985 m
T=beam thickness	0.003175 m
W=beam width	0.0762 m
D=distance between each beam	0.05715 m
M_T=arm total mass	4.8565 kg
E=modulus of elasticity	2.0 e11 N-m^2
ρ =density/unit volume	7861.05 kg/m^3

power unit shown in Figure 3.3., a Bird-Johnson 3-axis Hyd-Ro-Wrist, a Moog 760-100 servovalve, and associated Moog servocontroller and high pressure filter assembly.

The York hydraulic power unit includes a three horsepower motor and starter and is capable of supplying a system pressure of 2250 psi. Standard supply pressure for all runs was 2000 psi.

The actuator selected was a Bird-Johnson 3-axis Hyd-Ro-Wrist with a displacement in the pitch axis of 4.0 in^3 / rad. It is a hydraulic rotary actuator supported on roller bearings with 180 degrees of rotation. It was chosen by Petroka [Ref. 12] after a careful selection process. The principal consideration was a

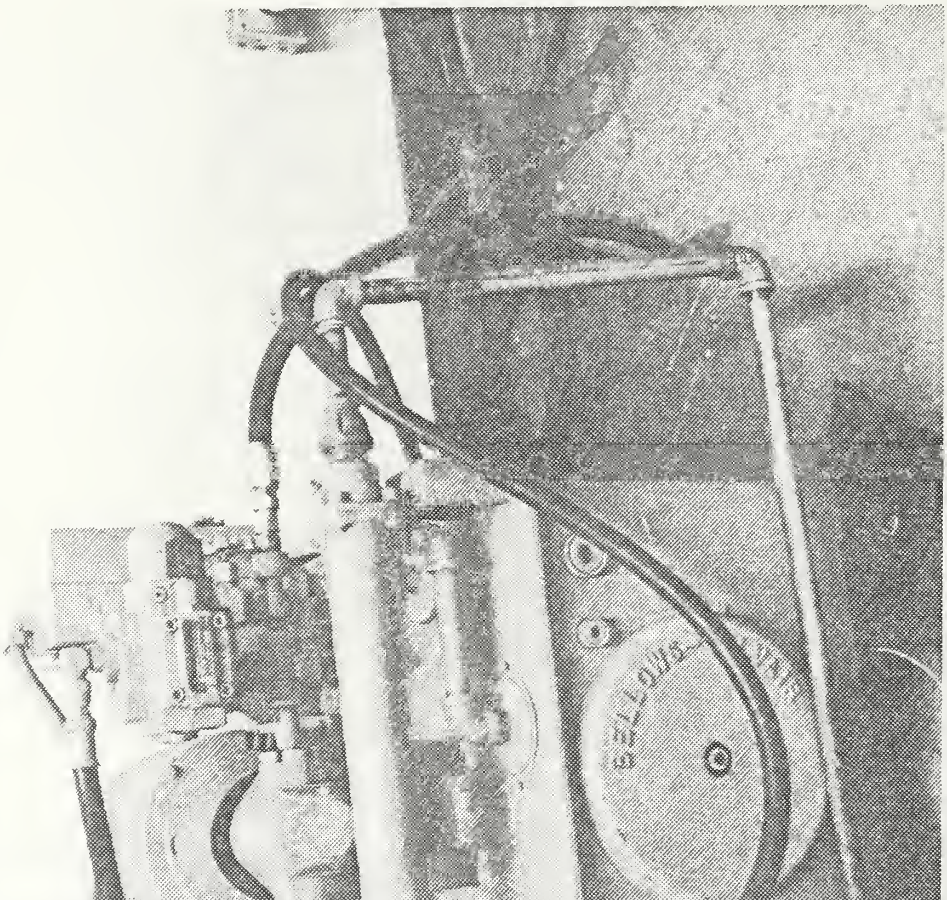

Figure 3.3. Hydraulic Power Unit

minimum displacement of 1.0788 in^3 / rad, based on an analysis of the system using the relationship

$$D_m = T_d / \eta_t P_L (2-31)$$

The wrist also includes roll and yaw axis, but these were disconnected and not used for this thesis.

The Moog 760-100 servovalve has a rated flow of 1.0 gpm. This was chosen to satisfy the design flow rate of 0.637 gpm. The 760 series servovalve is a two-stage design. The output stage is a closed center, four-way, sliding spool. The pilot stage is a symmetrical double-nozzle and flapper, driven by a double air gap, dry torque motor in a sealed compartment. Spool position feedback is provided by a cantilever spring.

3. Sensors

The flexible manipulator had three principal sensors- an accelerometer for acceleration , strain gages for strain, and a potentiometer for large motion angle.

The accelerometer was an Endevco model 7265A high sensitivity, small size (6 grams), damped, piezoresistive accelerometer. A piezoresistive type accelerometer was chosen for its good low frequency response, an important consideration in our case. The other predominant factor was small size for minimum mass loading. The damping helps to extend the useful high frequency range and reduce the effects of spurious high frequency excitation. The accelerometer was mounted at the tip of the flexible manipulator, with the axis of maximum response aligned along the length of the arm. Signal conditioning was provided by a BAM-1 amplifier.

Four CEA-06-12UW-350 strain gages were attached to the flexible manipulator. Two were located at the base and two at the mid-section of each steel bar, on opposite sides of the neutral axis. A mid-length strain gage was used for all data runs since it provided the maximum response. Signal conditioning was again provided by the BAM-1 amplifier.

A Bourn potentiometer was used to obtain large motion angle data. This was essential information for determining tip position in terms of the global coordinate system.

The heart of the data acquisition system was a Apple Macintosh 512K computer coupled with a MacADIOS analog-to-digital/digital-to-analog convertor. The MacADIOS is produced by GWI Insruments and the hardware is called a Model 411 Hardware Unit. It comes with its own software and serial interface to the Macintosh. It has 16 digital inputs and outputs, 8 analog inputs and 4 analog outputs, a programmable clock signal and a programmable counter. The MacADIOS can be programmed using Microsoft BASIC or C languages.

The strenth of the MacADIOS system is the wide array of software that is included with the system. This software includes the 5 Instruments Package, MacADIOS Manager, XY MaCorder, Microsoft BASIC library calls and source code for Manx Aztec C. The 5 Instruments Package permits the system to emulate a 8 input digital voltmeter, oscilliscope, spectrum analyzer, sonogram, and a spectrogram.

The software primarily used for the thesis was the portion called the MacADIOS Manager (MM) and the library of associated BASIC routines. This allowed the formulation of two experiments, one for tip position determination using accelerometer information, and the other for determining experimental strain from strain gage data . Two channels of analog data are input (accelerometer output and potentiometer output, or strain gage output and potentiometer output), this limits sampling to a 172 Hz rate or 5.81 milliseconds per sample [Ref. 15], the resulting waveforms are digitally stored awaiting further processing. Single channel inputs can be sampled at the much higher rate of 96 microseconds. The raw data can be graphically displayed using MM.

B. EXPERIMENTAL PROCEDURE

1. Tip Position Determination

The hydraulic power unit was started and allowed to run for approximately 30 minutes to allow the system temperature to stabilize. During this time the MacADIOS was set up to perform as a voltmeter so that initial calibration of the amplifier and accelerometer output could be done. Voltage to the servovalve controller was set at 2 volts. This produced an input current of 4 milliamps to the servovalve. This current was used to allow direct comparison with Petroka's [Ref. 12] results. The MacADIOS Manager was setup to sample two analog waveforms, accelerometer output and potentiometer output, at a rate of 6.00 milliseconds, or 166.67 Hz, and store these waveforms in digital form. This formatted the data for later transportation via modem to the IBM 3033 mainframe computer for data reduction. The manipulator was then loaded by attaching weights to the 4 studs at the tip of the manipulator. No load , 2.115 Kg, and 4.223 Kg load conditions were investigated. Once the load was attached to the manipulator, the manipulator was placed in the horizontal position and allowed to come to rest. From this position a step input of 4 milliamps was applied to the servovalve. Data were then collected by the MacADIOS.

2. Experimental Strain Determination

The procedure for experimental strain determination was very similiar to the procedure for tip position determination described above. The major differences being that strain gage output is taken vice accelerometer output and the flexible manipulator is initially in the vertical position so that the initial strain in 0.0 micrometers/meter.

IV. RESULTS

The experimental work was divided into three areas. First a comparison was made between the two methods of determining tip position, accelerometer and motion pictures. These results were used as a baseline to compare this thesis with past work. Then a comparison was made between experimental tip position and tip position determined from the two simulation programs one using a modal shape function the other using a cubic shape function. Finally a comparison was made between experimental and theoretical strain again comparing the two theoretical models to the experimental value.

A. ACCELEROMETER VERSUS MOTION PICTURES

One of the primary objectives of this research was to investigate a different technique for determining arm tip position. Petroka [Ref. 12] considered several methods for his thesis including motion pictures, an automatic laser tracking interferometer system, digitizing tablets, ultrasonics, position/displacement transducers, digitizing vision systems and accelerometers before finally choosing motion pictures. The principal factors in the decision were simplicity, availability, and cost. He encountered some limitations to this method during the course of his thesis. These included requirement for a slow, tedious frame-by-frame examination hindered by the lack of clarity of the arm tip position and by the absence of definition of the background grid measurement. This method is also limited in that it cannot be used to feedback position data to a controller for appropriate control action. .

The additional method investigated in this thesis was the use of an accelerometer. A piezoresistive accelerometer was attached to the end of the flexible arm. A potentiometer was attached to the rotary hydraulic actuator at the base of the flexible manipulator. Accelerometer output and potentiometer output were then used to obtain tip position information. This required two integrations and a global transformation. The accelerometer senses a global acceleration we will call $\ddot{S}$. This can be related to X and Y global coordinates. Using our global transformation matrix we can then obtain an expression for $\ddot{v}(0)$. This is then integrated twice using a DSL simulation program. Once again using the global transformation we can obtain an expression for tip position in terms of the global coordinate system. This process is discussed in further detail in Appendix C. Figures 4.1-4.6 show the comparisons between tip position determined by Petroka using motion pictures and by accelerometer in this thesis. The three loading conditions were no load, a 2.115 kg load, and a 4.233 kg load. For comparative purposes, a percent difference was calculated with motion picture data representing the reference. This is given in Table 4.1. All experimental runs were made with a 4 milliamp step input of current. The initial condition for all runs was the horizontal position of the flexible arm.

The figures and table values indicate that there is good agreement between the two methods for determining tip position. There is closer agreement as the loading increases. This is most probably because the lower speeds at the higher loads allow more accurate reading of the motion picture film. Use of an accelerometer would seem to be the preferred method since the accuracies are comparable and the accelerometer output does not require the tedious, manual interpretation that is required of the motion film data.

TABLE 4.1

PERCENT DIFFERENCE BETWEEN EXPERIMENTAL TECHNIQUES

LOAD	X	Y
0 Kg	1.2	15
2.115 Kg	0.50	9.3
4.233 Kg	0.44	6.2

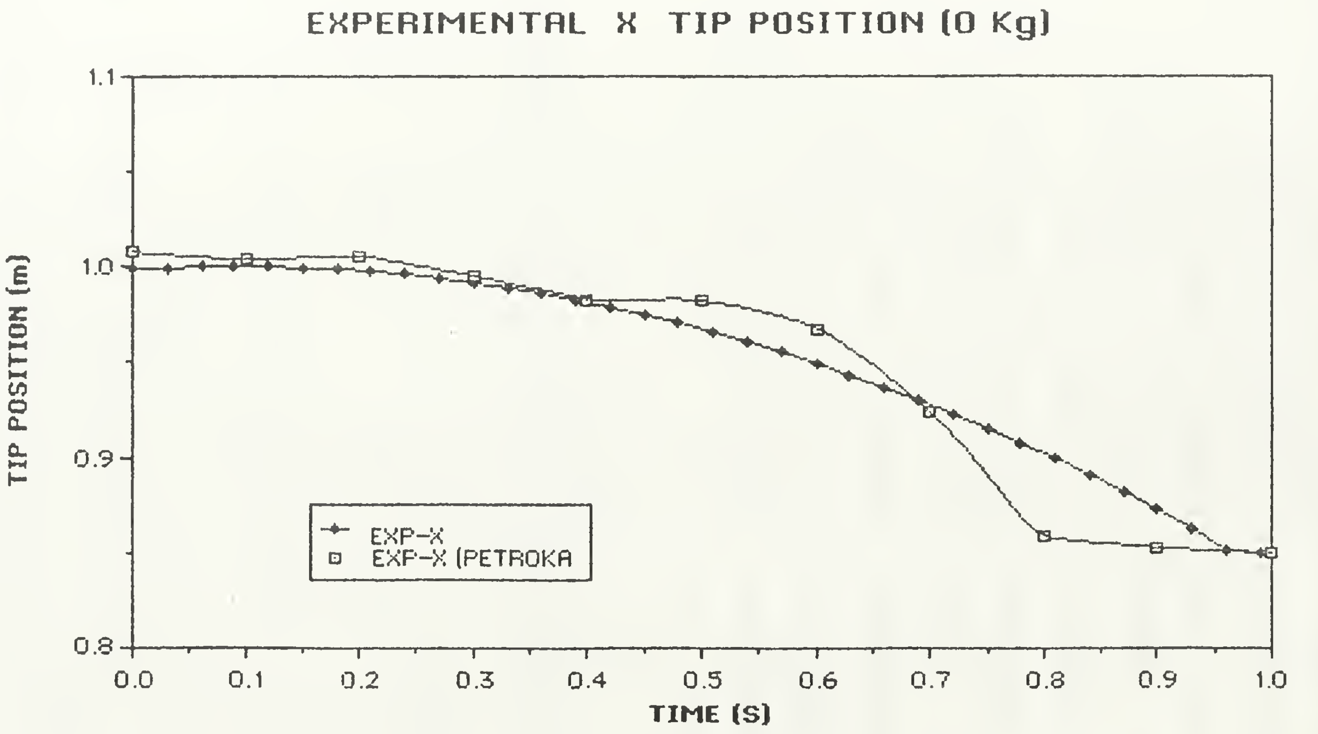

Figure 4.1. Comparison Between Experimental Techniques. (X, 0 Kg)

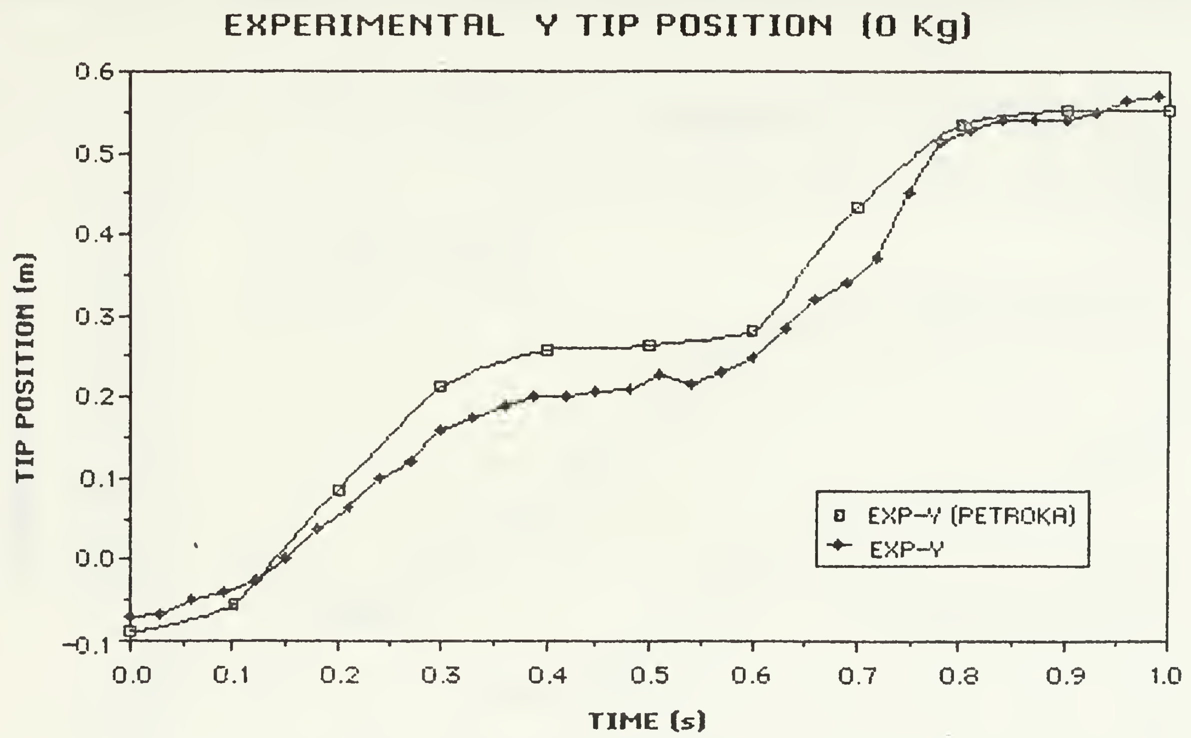

Figure 4.2. Comparison Between Experimental Techniques. (Y, 0 Kg)

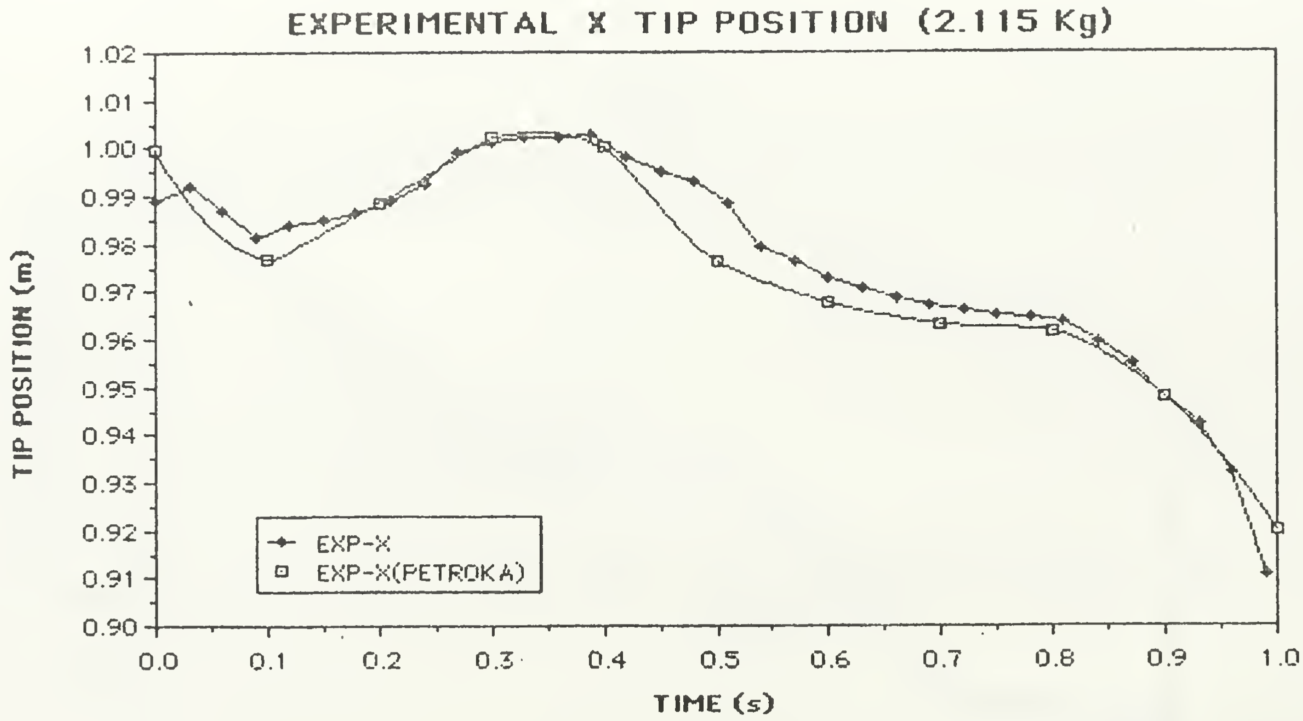

Figure 4.3. Comparison Between Experimental Techniques. (X, 2.115 Kg)

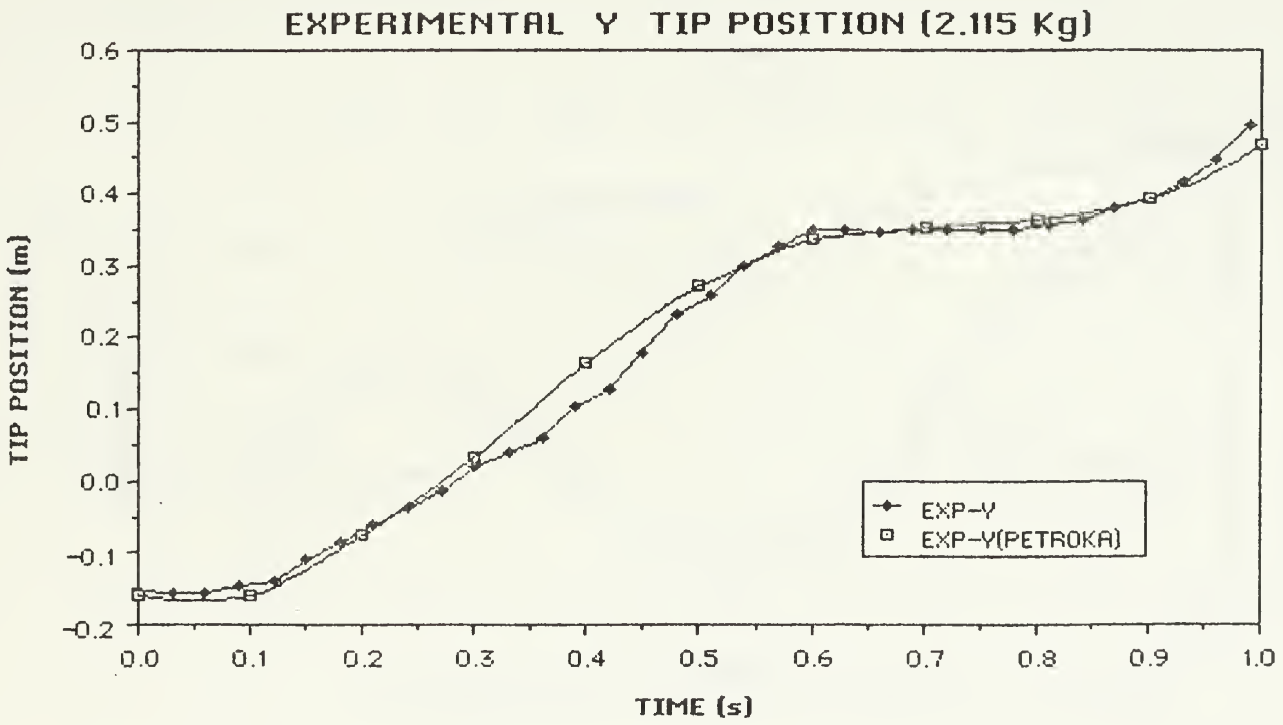

Figure 4.4. Comparison Between Experimental Techniques. (Y, 2.115 Kg)

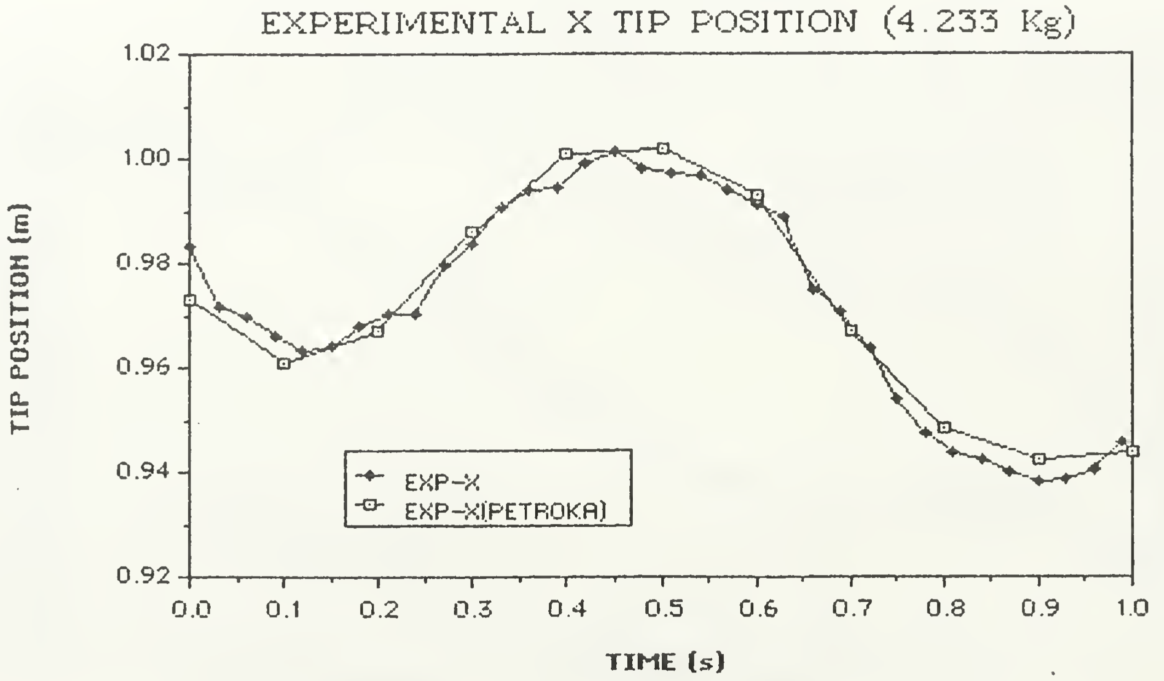

Figure 4.5. Comparison Between Experimental Techniques. (X, 4.233 Kg)

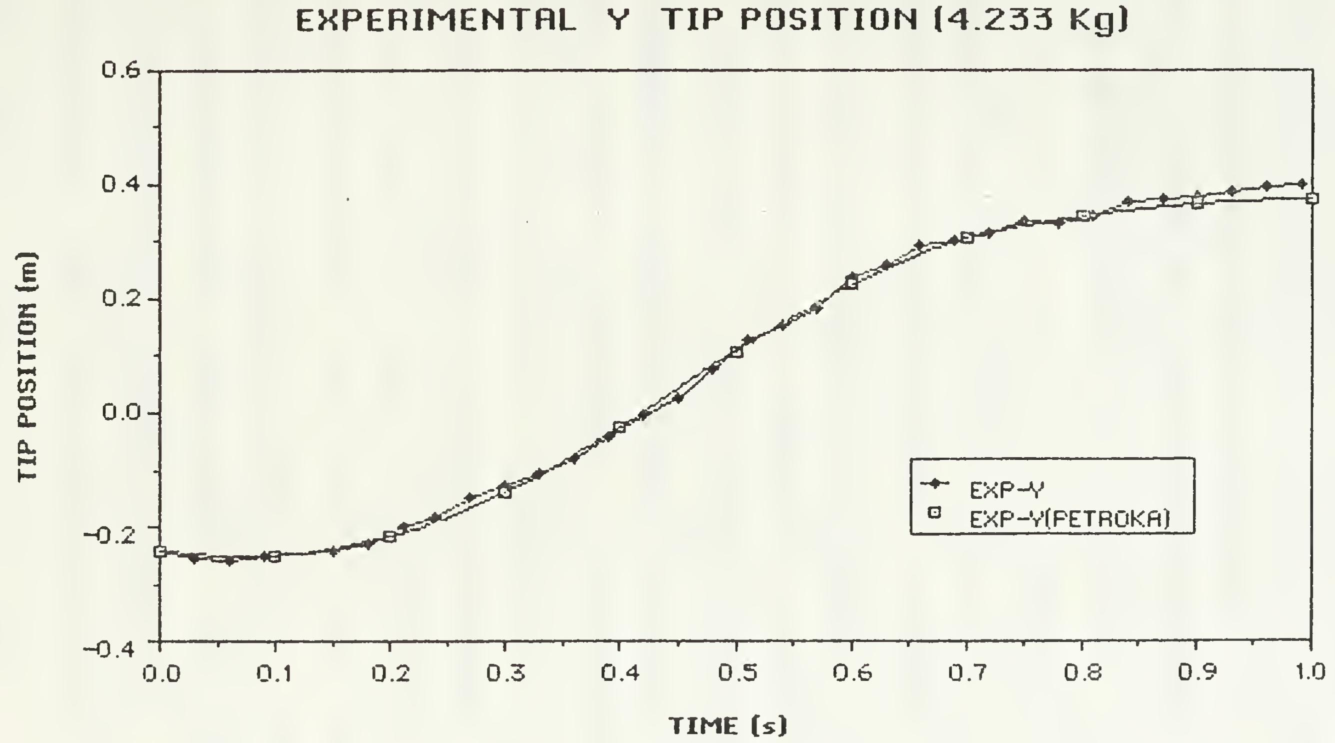

Figure 4.6. Comparison Between Experimental Techniques. (Y, 4.233 Kg)

B. TIP POSITION

After evaluating the different techniques for determining arm tip position the next step in the validation of the ERLS dynamic model was to compare the actual arm tip position as determined by accelerometer with the arm tip position predicted by the two different simulation models.

Evaluation of the data centered around the parameters amplitude and frequency. These were evaluated in Petroka's thesis [Ref.12] and for a consistent approach in validating the ERLS model are evaluated in this thesis. All criteria for comparison were also adopted from Petroka's thesis. These include a relative percentage error of +/- 10% as the standard for comparison. The tip position amplitude error is computed by taking the square root of the sum of the squares of the differences between the theoretical and experimental X and Y coordinate positions. The first mode tip frequency errors are computed by taking the difference between the theoretical and experimental tip position frequencies. Only the first mode frequency is determined since it was the dominant mode. Errors were then normalized to obtain relative errors. Tip position amplitude was normalized using arm length, and frequencies were normalized using experimental frequencies.

Figures 4.7-4.12 represent the graphical results of the comparison of experimental arm tip position with theoretical arm tip position. Results are given in terms of the global coordinate system. The loading conditions were no load, a 2.115 kg load and a 4.233 kg load. The initial condition for all runs was the horizontal position of the flexible arm. Servovalve excitation for all runs was a step input of 4.0 milliamps. The initial tip deformation and slope as determined by elementary beam theory were initial conditions input into the simulation program.

The figures and results tabulated in Tables 4.2 and 4.3 indicate that both models satisfy the evaluation criteria of 10 % error. The modal shape function gives us slightly better results percentage wise with regards to tip position amplitude, but when converted to actual distances these small percentage differences can mean significant improvements in tip position accuracy since a 1 % improvement is on the order of a 1 centimeter improvement in accuracy. The acceptable tip position amplitude error occurs despite the fact that we are using a truncated model of only the first two modes.

The differences in amplitude observed in the arm tip position amplitude can attributed to several causes. Experimentally there are errors associated with recording the data such as calibration errors and conversion errors. The cubic shape function is limited by the single element modeling which will result in a more rigid model. The modal shape function model suffers from the

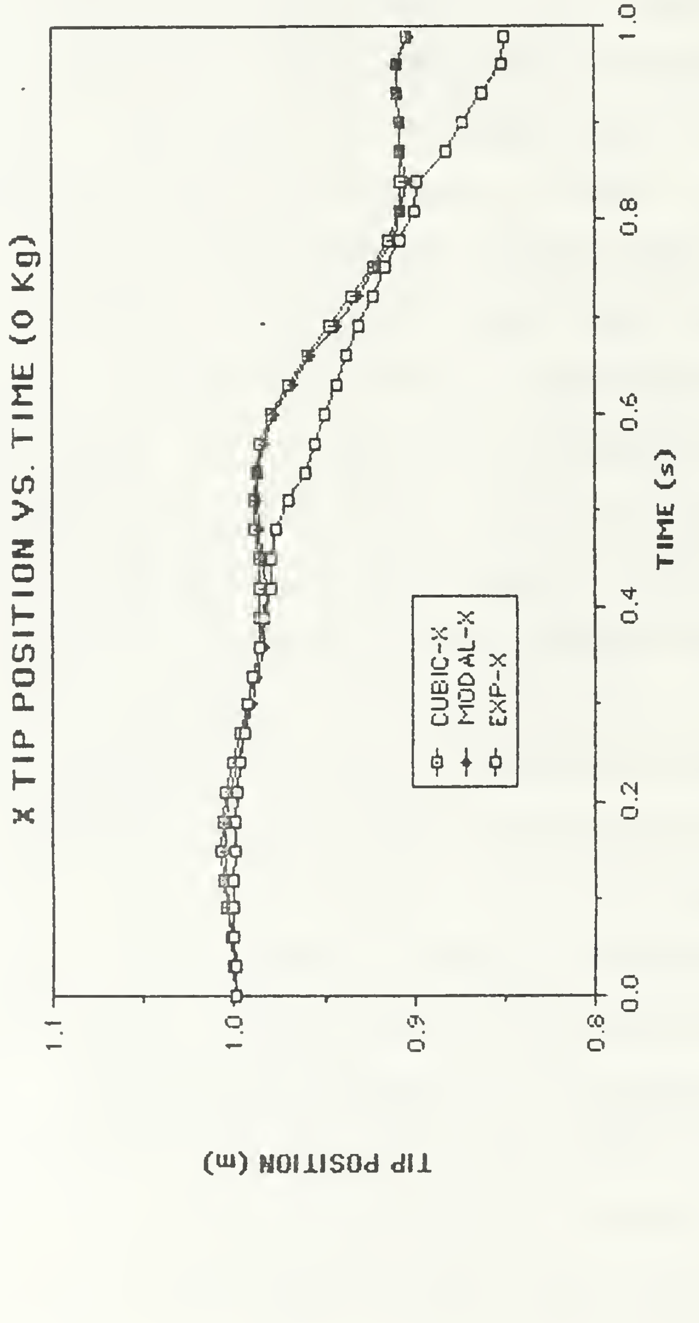

Figure 4.7. Tip Position. (X, 0 Kg)

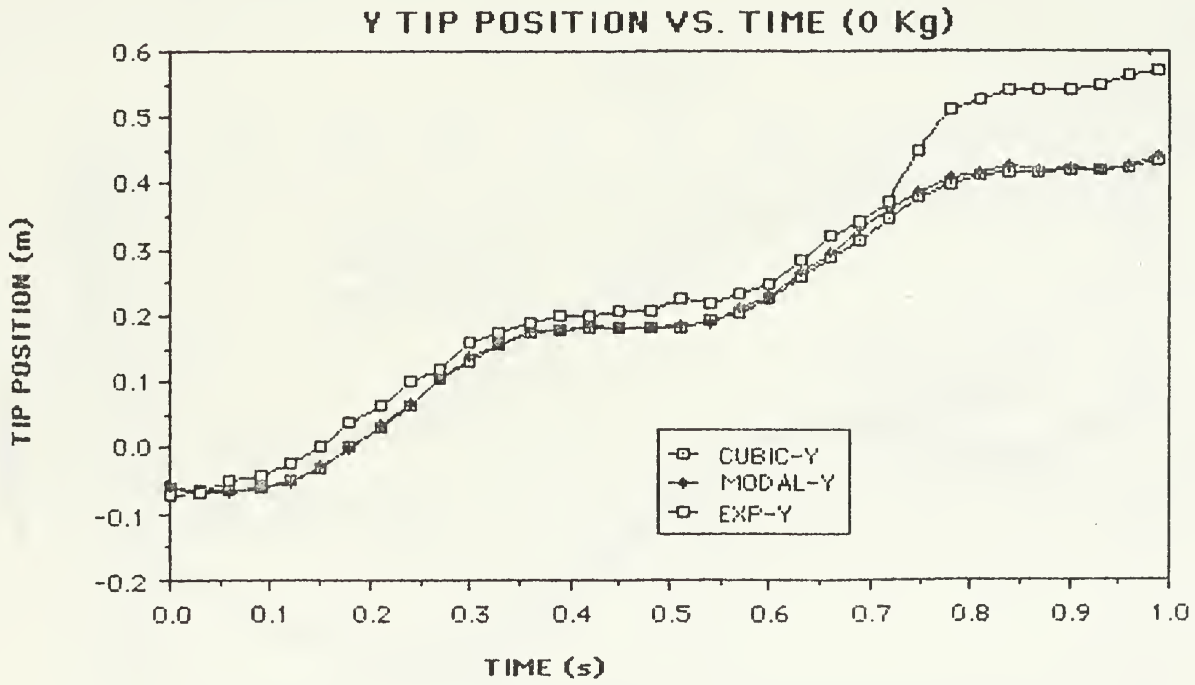

Figure 4.8 Tip Position (Y, 0 Kg)

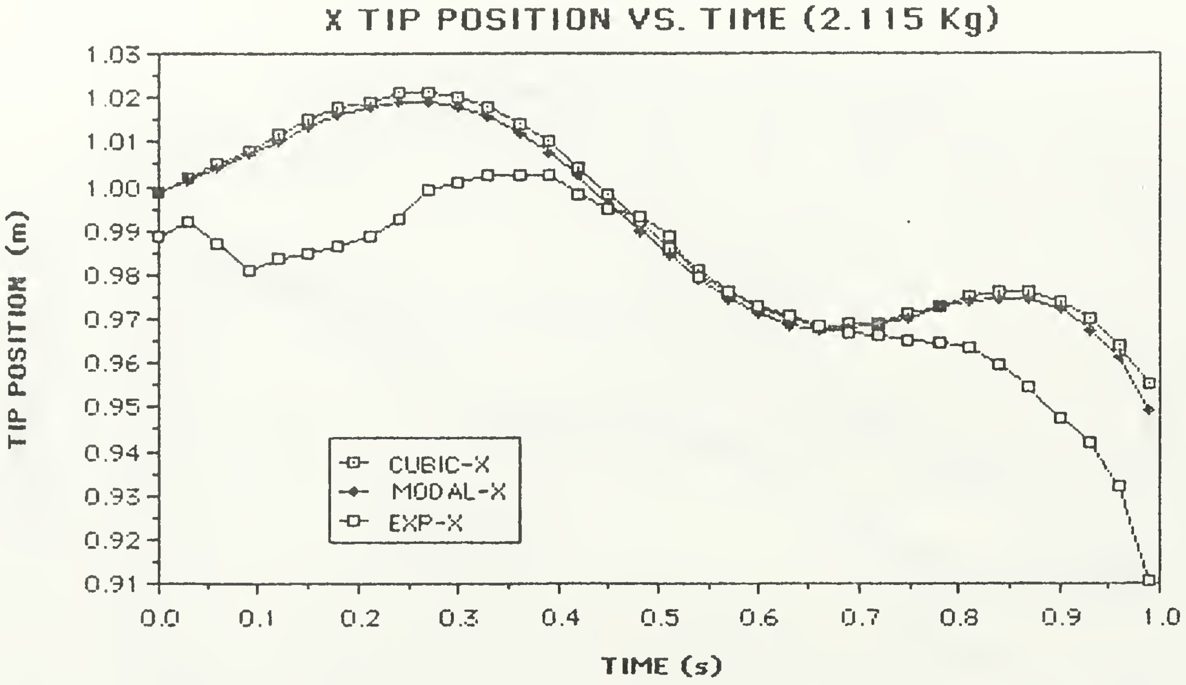

Figure 4.9. Tip Position (X, 2.115 Kg)

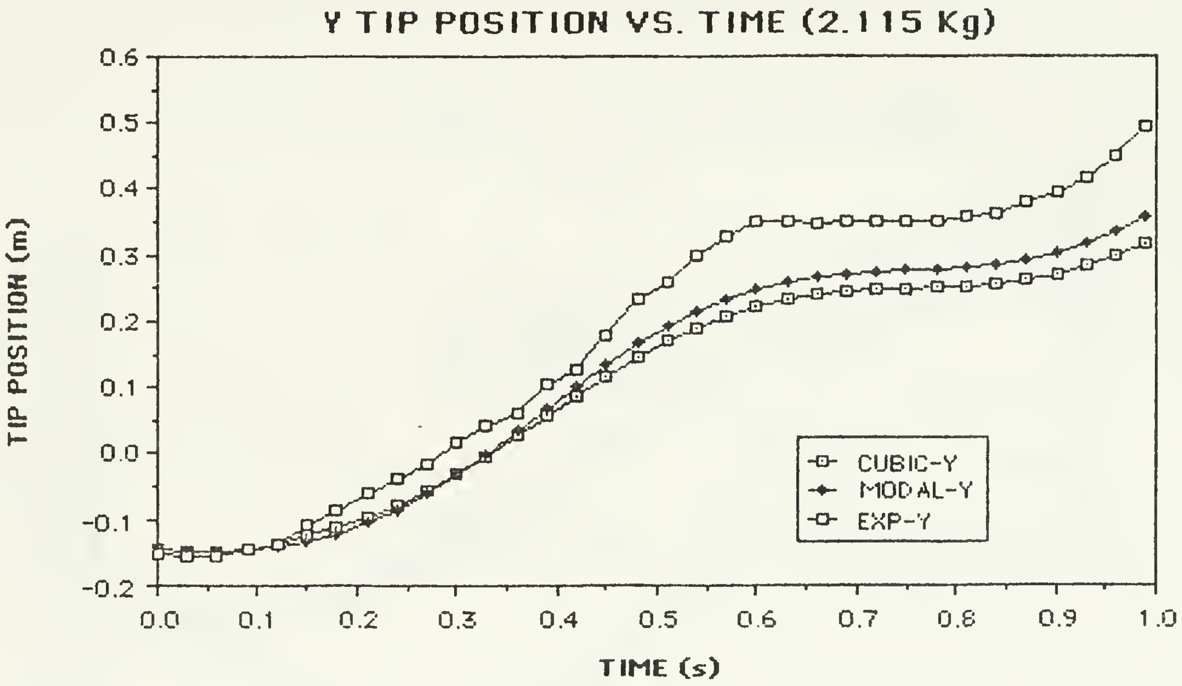

Figure 4.10. Tip Position (Y, 2.115 Kg)

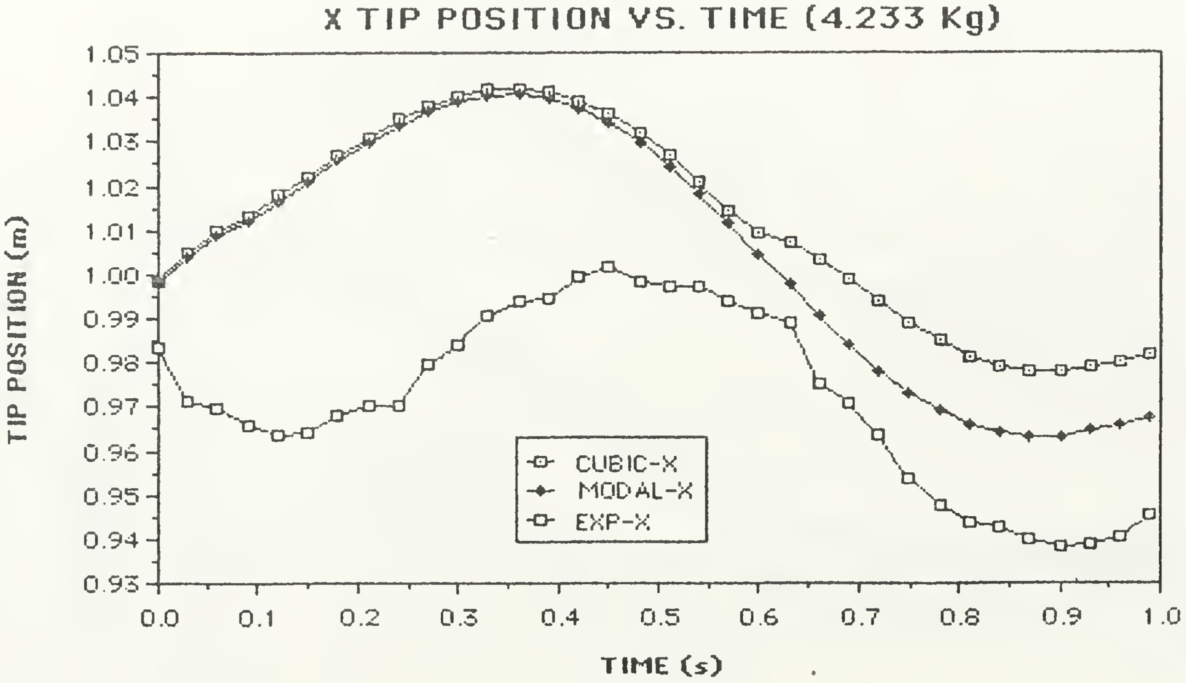

Figure 4.11. Tip Position. (X, 4.233 Kg)

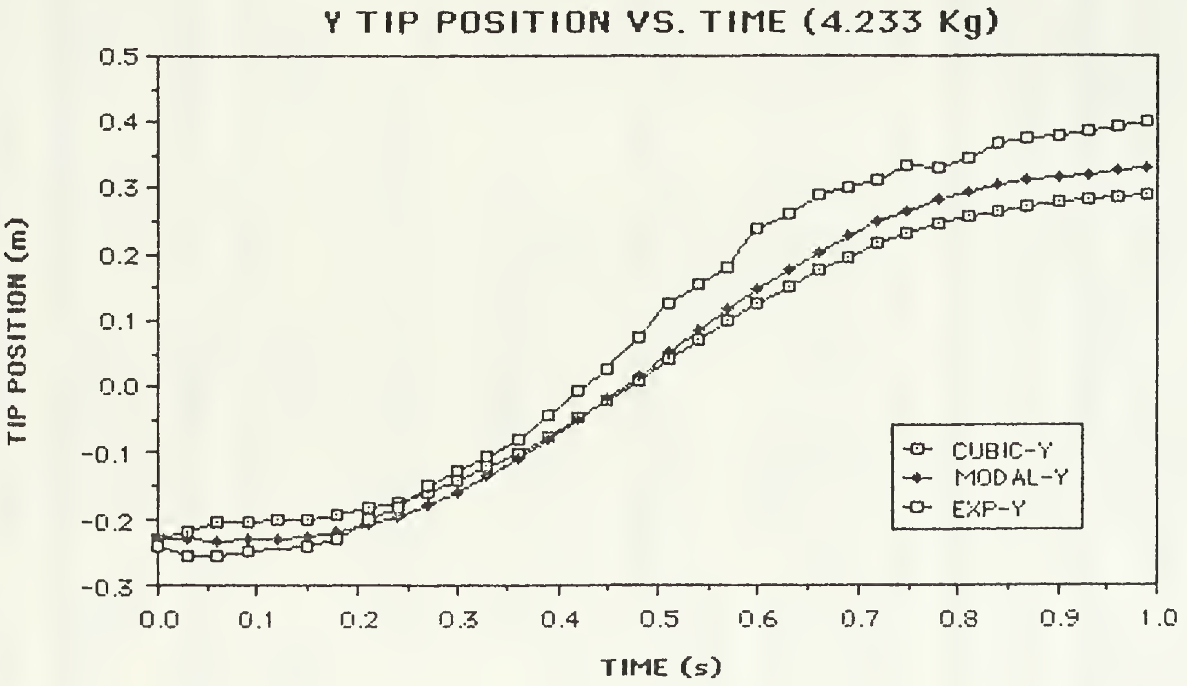

Figure 4.12. Tip Position (Y, 4.233 Kg)

TABLE 4.2

RELATIVE ERRORS OF TIP POSITION AMPLITUDE		
LOAD	CUBIC	MODAL
0 Kg	5.3 %	4.9 %
2.115 Kg	7.9 %	6.3 %
4.233 Kg	8.3 %	6.3 %

TABLE 4.3

RELATIVE ERROR OF FREQUENCY		
LOAD	CUBIC	MODAL
0 Kg	< 5 %	< 5 %
2.115 Kg	< 5 %	< 5 %
4.233 Kg	< 5 %	< 5 %

truncation effects since the flexible arm is theoretically composed of and infinite number of modes and we have only include the first two modes. Neglecting axial deformation and assuming small bending displacements also effects the accuracy of our model. These effects are most noticeable during the first few tenths of a second of motion. During this time arm tip position decreases but an increase in predicted by both the cubic and modal shape functions. The theoretical arm position approximates the actual arm position but since the the small motion displacement is measured with respect to the ERLS local coordinate system the theoretical arm length appears to increase particularly under heavy loading.

The frequency results were similarly promising. All met the error criteria of 10 %. Only the fundamental frequencies were observed indicating the dominance of the first mode.

C. STRAIN

The final step in the experimental validation of the ERLS model of a flexible manipulator was to compare experimental strain with the strain predicted from the two simulation models.

Amplitude and frequency were evaluated as in the tip position experiments. In addition, the error criteria of 10 % was also maintained. The strain amplitude and frequency errors were obtained by taking the difference between the theoretical and experimental strain data. The results are normalized using the respective experimental values. The strain amplitude is actually a superposition of the first two modes but the fundamental mode is dominant. The strain frequency is easily separated into fundamental and secondary modes.

Figures 4.13-4.15 are the graphical results of the comparison between experimental strain and theoretical strain as predicted by the ERLS using a a modal shape function. The loading conditions were no load, a 2.115 kg load and a 4.233 kg load. The initial condition for all runs was the vertical position of the flexible arm. Servovalve excitation for all runs was a step input of 4.0 milliamps. These results are summarized in Tables 4.4 and 4.5.

The strain amplitude values exeed the 10 % error criteria. The amplitude of the theoretical strain is usually less than the experimental values. This is to be expected from the truncation of the modal shape function. The addition of more modes to the model should bring the amplitude closer to the experimental values. The values predicted by the modal shape function are much closer than the cubic shape function results predicted in Petroka's thesis [Ref. 12], where errors were on the order of 35-40 %.

Perhaps the most significant result in the choice of a modal shape function over a cubic shape function is in the area of strain frequency. Fundamental frequencies are accurately modeled by both models with all results less than 5 %. However the modeling of the secondary mode frequency is much more accurately made with the modal shape function. The cubic shape function results in errors on the order of 20-40 % [Ref 12]. The modal shape function improves on these to the point where all errors satisfy the 10 % criteria. The theoretical values are all slightly greater than the experimental values, indicative of the more rigid model that results from the truncation of the modal shape function.

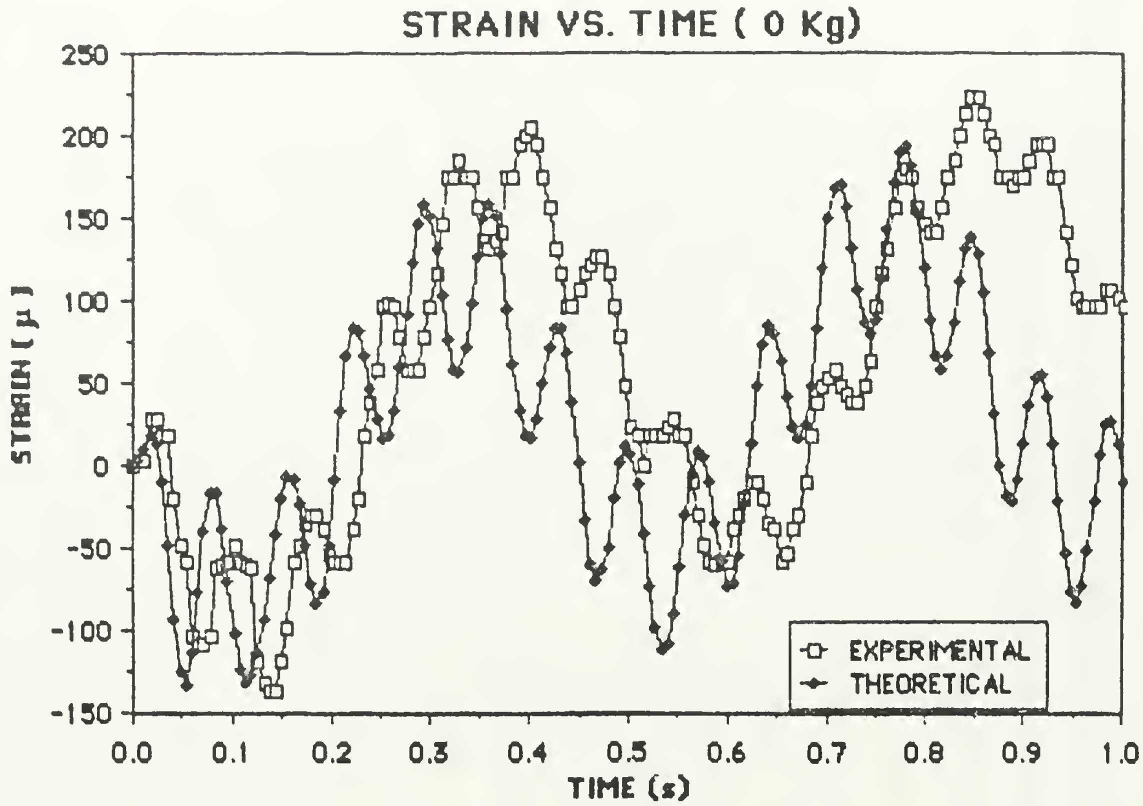

Figure 4.13. Strain (0 Kg)

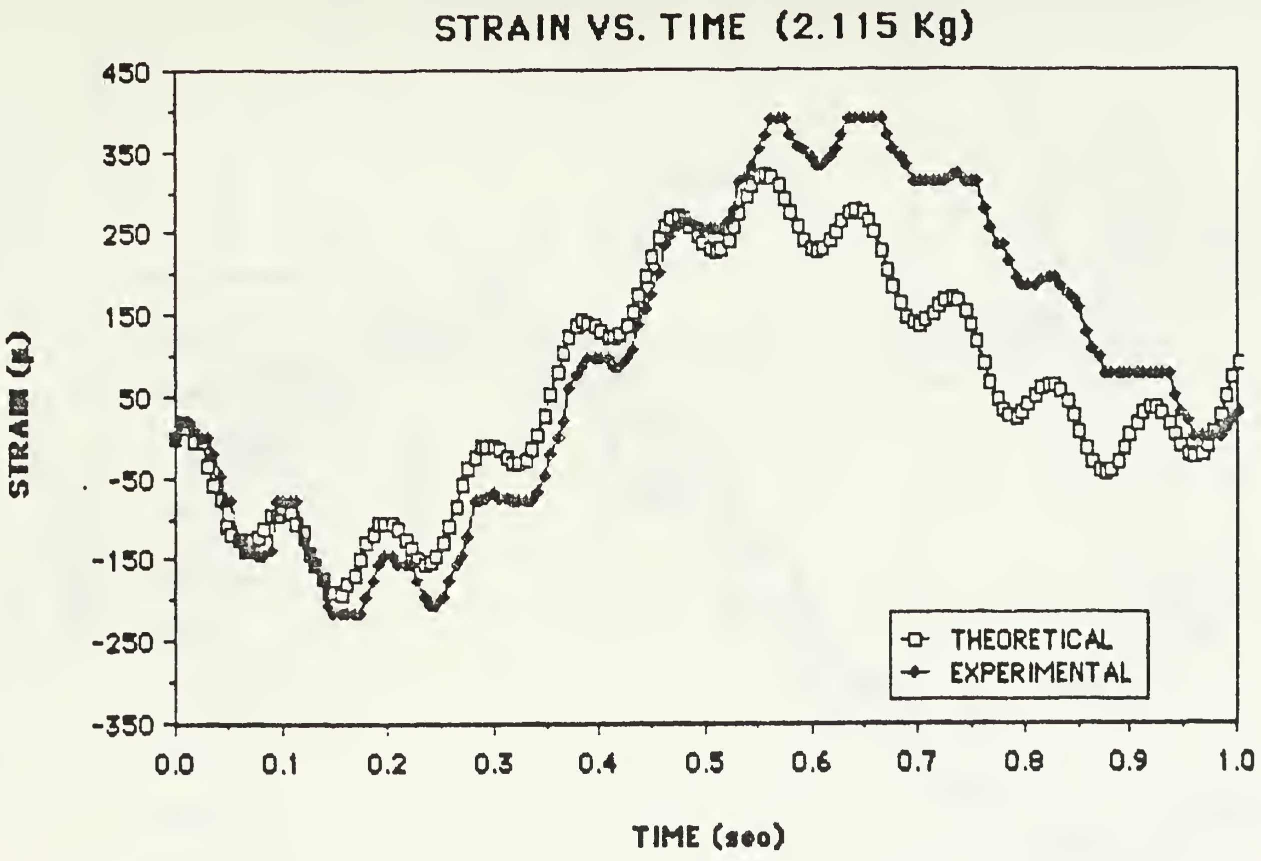

Figure 4.14. Strain (2.115 Kg)

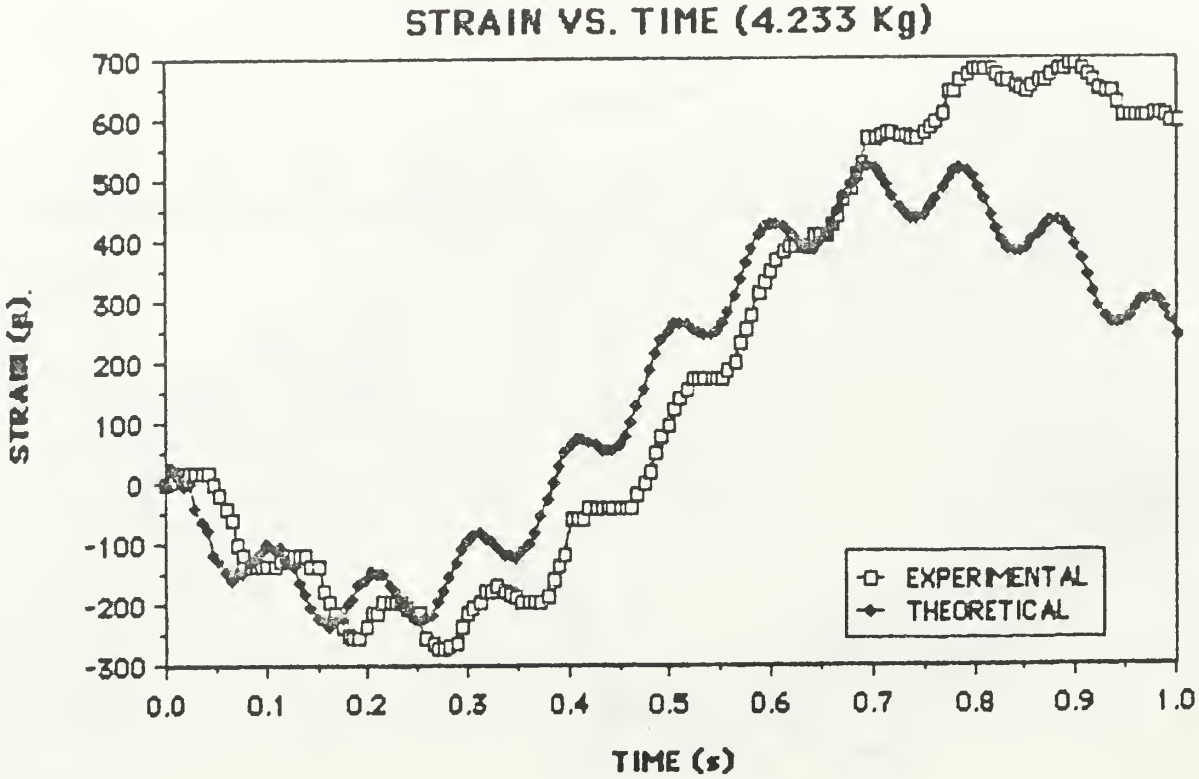

Figure 4.15. Strain (4.233 Kg)

TABLE 4.4

<table>
<tr><th colspan="3">RELATIVE ERROR OF STRAIN AMPLITUDE</th></tr>
<tr><th>LOAD</th><th>CUBIC</th><th>MODAL</th></tr>
<tr><td>0 Kg</td><td>38 %</td><td>17.9 %</td></tr>
<tr><td>2.115 Kg</td><td>38 %</td><td>17.4 %</td></tr>
<tr><td>4.233 Kg</td><td>41 %</td><td>23.7 %</td></tr>
</table>

TABLE 4.5

<table>
<tr><th colspan="5">RELATIVE ERROR OF FREQUENCY</th></tr>
<tr><th rowspan="2">LOAD</th><th colspan="2">FUNDAMENTAL</th><th colspan="2">SECOND MODE</th></tr>
<tr><th>CUBIC</th><th>MODAL</th><th>CUBIC</th><th>MODAL</th></tr>
<tr><td>0 Kg</td><td colspan="2">< 5 %</td><td>38 %</td><td>7.5 %</td></tr>
<tr><td>2.115 Kg</td><td colspan="2">< 5 %</td><td>26 %</td><td>< 5 %</td></tr>
<tr><td>4.233 Kg</td><td colspan="2">< 5 %</td><td>22 %</td><td>5.5 %</td></tr>
</table>

V. SUMMARY

A. CONCLUSIONS

The purpose of this thesis is to experimentally validate the ERLS dynamic model of a flexible manipulator. This validation is an essential part of the long-term goal of designing and controlling a flexible manipulator. The model chosen was Chang's [Ref. 5] Equivalent Rigid Link System dynamic model. The model was adapted to our experimental manipulator, a single-link, hydraulically actuated flexible arm moving in the vertical plane. The thesis investigated experimental techniques and a comparison of tip position amplitude and strain for the two simulation models.

The results indicate that an accelerometer and motion picture studies of arm tip motion yield comparable results. The accelerometer has the advantage in that it provides a continuous signal and does not require the tedious manual interpretation that is required of the motion picture data. This means an appreciable savings of time.

The tip position amplitude comparison between the experimental results and the results obtained from the two choices of shape function indicated that the choice of natural modes yields a slight percentage improvement in tip position accuracy, but this improvement can be significant in terms of distance particularly with regard to future control designs. Both shape functions accurately modeled the dominant first mode frequency of the arm tip position.

The comparison between experimental strain and the two theoretical simulations demonstrated that the natural mode shape function leads to significant improvements in both strain amplitude and frequency.

The limitations of both models are a direct result of some modeling assumptions and choices. The natural mode shape function is only composed of the first two modes. This truncation accounts for the lower values for both tip position and strain amplitude. This also yields a more rigid model, thus the slightly higher frequencies for the modal shape function. Similarly, the cubic shape function is restricted by its single element modeling. This more rigid representation accounts for the cubic shape functions lower amplitudes and higher freqeuncies.

The small displacement assumption leads to noticeable differences in tip position amplitude particularly during the first few tenths of a second of motion in the X direction. No effects of the small displacement assumtion are noted in the strain results.

The experimental and simulation results provided a valuable validation of the ERLS model. The choices of a natural mode shape function and an accelerometer for tip position determination offer improvements to the model and further suggest the potential benefits of the ERLS as a model for flexible manipulator.

B. RECOMMENDATIONS

The long-term goal of this research is the design and control of a flexible manipulator system. Simulation studies need to continue on the validation of the ERLS model. Shear deformation and rotary inertia effects should be considered in future work. In addition, the effects of truncation also need to be studied. The significance of each of the terms in the equations of motion would be helpful in improving the model.

Control system design requires significant investigation. Specific areas of importance include control schemes to achieve motion control and force control. Continued studies must be made in the area of finding sensors that will provide accurate and fast feedback signals. Hardware implementations to make eventual on-line control possible should also be investigated.

APPENDIX A

DERIVATION OF THE EQUATIONS OF MOTION FOR THE EXPERIMENTAL, SINGLE-LINK, FLEXIBLE MANIPULATOR

A principal idea of the ERLS is to separate the motion of a flexible manipulator system into a large motion and a small motion. The large motion represents the the global motion of the system described by the ERLS and the small motion displacements are described with respect to the ERLS. The Lagrangian dynamics approach is used for the derivation of the equations of motion due to its straightforward and systematic nature which is helpful in the analysis of complex systems. This formulation requires generalized coordinates. Since the global motions are separated into large motions and small motions, generalized coordinates are defined to represent these two motions. The three generalized coordinates chosen are the rigid body rotation, θ, and two nodal displacements, $v(0)$ and $\phi(0)$, which can be measured at the end of the link. The following are the two sets of Lagrange equations used to develop the equations of motion:

$$d/dt\,(\partial KE/\partial\theta) - \partial KE/\partial\theta + \partial PE/\partial\theta = F$$

$$d/dt\,(\partial KE/\partial U) - \partial KE/\partial U + \partial PE/\partial U = 0$$

KE - kinetic energy

PE - potential energy

θ - large motion joint variable, theta, measured between the ERLS link and the global X axis

U - 2 x 1 vector of small motion displacement and slope, $v(0)$ and $\phi(0)$

F - generalized force for large motion, applied moment

Kinetic energy of the system is due to kinetic energy of the link, kinetic energy of the actuator and the kinetic energy of any load. The actuator is treated as a rigid body and all kinetic energy terms are calculated separately. Generalized forces include any applied forces and damping forces. The expressions used for the determination of kinetic energy are as follows:

$$\text{KE (arm)} - 1/2 \int_{\substack{\text{ARM}\\ \text{VOLUME}}} \mu \, \dot{R}^T (\dot{R}) \, dv$$

$$\text{KE (load)} - 1/2 \text{ Trace} \int_{\substack{\text{LOAD}\\ \text{VOLUME}}} \mu_L \, \dot{R}_L{}^T (\dot{R}_L) \, dv$$

$$\text{KE (rotor)} - 1/2 \text{ Trace} \int_{\substack{\text{ROTOR}\\ \text{VOLUME}}} \mu_R \, \dot{R}_R{}^T (\dot{R}_R) \, dv$$

The global position vector of the arm is determined form the following transformation:

$$R = W\,(r + d)$$

W = the 3 x 3 transformation matrix , function of theta only

r = the 3 x 1 local position vector of the arm measured from the coordinate system whose origin is at the end of the ERLS link

d = the 3 x 1 deformation vector that only includes the transverse displacement, v. Derivation of this vector is made in Chapter 2.

μ = the mass density of the steel arm

The global position vector of the load is determined from the following transformation:

$$R_L = W\,D_L\,r_L$$

D_L = the 3 x 3 transformation matrix due to the local deformations of the arm tip

r_L = the 3 x 1 local position vector for the load

μ_L = mass density of the load, steel

The global position vector of the hydraulic actuator is given by the following transformation:

$$R_R = A_R\,r_R$$

A_R = the 3 x 3 transformation matrix due to the large motion rotation of the rotor

r_R = the 3 x 1 local position vector for the rotor

μ_R = the mass density of the actuator rotor, aluminum

Potential energy of the system comes from strain energy due to deformation and from gravity. The expressions used for the determination of potential energy are as follows:

$$PE_d = 1/2 \int_{\substack{\text{LINK}\\\text{LENGTH}}} E\, I_{zz}\, (v'')^2 \, dx$$

$$PE_g = - \int_{\substack{\text{LINK}\\\text{VOLUME}}} \mu \; r^T g \; dv$$

$$PE_{gL} = - \int_{\substack{\text{LOAD}\\\text{VOLUME}}} \mu_L \; r_L^{\,T} g \; dv$$

$E\, I_{zz}$ = the flexural rigidity in the z direction, perpendicular to the plane of motion

v'' = second derivative of the transverse displacement v with respect to the the local x coordinate direction expressed in terms of v(0) and ϕ(0).

g = gravitational acceleration vector

To simplify the expressions in the equations of motion the following definitions for inertia terms, adapted from the expressions found in Reference 5, are used:

$$I_L = \int\limits_{\substack{\text{LOAD}\\ \text{VOLUME}}} \mu_L \, r_L r_L^T \, dv$$

$$= \text{ the 3 x 3 inertia matrix of the load}$$

$$I_R = \int\limits_{\substack{\text{ROTOR}\\ \text{VOLUME}}} \mu_R \, r_R r_R^T \, dv$$

$$= \text{ the 3 x 3 inertia matrix of the rotor}$$

$$I111(W_\theta^T \, , \, W_\theta) = \int\limits_{\substack{\text{LINK}\\ \text{VOLUME}}} \mu \; r^T \; W_\theta^T \; W_\theta \; r \, dv$$

$$I112(W_\theta^T \, , \, W) = \int\limits_{\substack{\text{LINK}\\ \text{VOLUME}}} \mu \; r^T \; W_\theta^T \; W \; \Phi \, dv$$

$$I121(W^T \, , \, W_\theta) = \int\limits_{\substack{\text{LINK}\\ \text{VOLUME}}} \mu \; \Phi^T \; W_\theta^T \; W \; r \, dv$$

$$I122(W^T \, , \, W) = \int\limits_{\substack{\text{LINK}\\ \text{VOLUME}}} \mu \; \Phi^T \; W^T \; W \; \Phi \, dv$$

$$I122(W^T\ ,\ \ddot{W}_R) = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu\ \Phi^T\ W^T\ \ddot{W}_R\ \Phi\ dv$$

$$I122(W_\theta{}^T\ ,\ \dot{W}) = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu\ \Phi^T\ W^T\ \dot{W}\ \Phi\ dv$$

$$I122(W_\theta{}^T\ ,\ W_\theta) = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu\ \Phi^T\ W_\theta{}^T\ W_\theta\ \Phi\ dv$$

$$I_{YY} = \int\limits_{\substack{\text{LOAD}\\\text{VOLUME}}} \mu_L\ y^2\ dv$$

$$I_{XX} = \int\limits_{\substack{\text{LOAD}\\\text{VOLUME}}} \mu_L\ x^2\ dv$$

Φ = the 3 x 2 link shape matrix defined in Chapter 2

W_θ = derivative of the 3 x 3 matrix W with respect to the joint variable theta

$\ddot{W}_R$ = result of the simplification of the second time derivative of the transformation matrix W and is termed the residual acceleration.

The following definitions are utilized to simplify the computations and the expressions used for computer coding of the equations of motion:

$$K_{11} = \int\limits_{\substack{\text{LINK}\\\text{LENGTH}}} \Gamma^T C \Gamma \; dx$$

$$H_{11} = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu \;\; r^T \; dv$$

$$H_{21} = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu_L \;\; \Phi^T \; dv$$

$$H_{41} = \int\limits_{\substack{\text{LINK}\\\text{VOLUME}}} \mu_L \;\; r_L{}^T \; dv$$

Γ = the second derivative of the shape function matrix

C = the 3 x 3 flexural rigidity matrix including only $E\,I_{22}$

The actual derivation of the equations of motion requires detailed, tedious substitution of the expressions for kinetic energy and potential energy into the Lagrange equations, and the Lagrangian formulation yields two sets of equations. One set describes the large motions and the other set describes the small motions. These two sets of equations are non-linear, coupled, second-order, ordinary differential equations represented as follows,

$$M_{qq}\ \ddot{\theta} + M_{q\eta}\ \ddot{U} = F_q \tag{2-1}$$

$$M_{\eta q}\ \ddot{\theta} + M_{\eta\eta}\ \ddot{U} + K_\eta\ U = F_\eta \tag{2-2}$$

where

M_{qq} = 1 × 1 effective inertia matrix for large motions

$M_{q\eta}$ = 1 × 2 coupled inertia matrix of the small motion effect on large motion

$M_{\eta q}$ = 2 × 1 coupled inertia matrix of the large motion effect on small motions

$M_{\eta\eta}$ = 2 × 2 effective inertia matrix for small motions

K_η = 2 × 2 stiffness matrix for small motions

F_q = 1 × 1 load vector for the large motions

F_η = 2 × 1 load vector for the small motion

$\ddot{\theta}$ = second derivative of the generalized coordinate of the large motions

U = 2 × 1 vector ,generalized coordinate of the deformations representing the small motions

The coefficients for the terms are defined as follows:

$$M_{qq} = \quad I111(W_\theta^T\ ,\ W_\theta) + U^T\ I122(W_\theta^T\ ,\ W_\theta)\ U \quad +$$

$$\mathrm{Trace}\ (\ W_\theta\ D_L\ I_L\ D_L^T\ W_\theta^T\ +\ A_R\ I_R\ A_R^T\ )$$

$$M_{q\eta}\ =\ \ I112(W_\theta^T\ ,\ W)\ +\ \ ((M_L\ L\ +\ M_x),\ (M_x\ L\ +\ I_{xx}\ +I_{yy})$$

L is the arm length. M_L is the mass of the load. M_x is the first moment of the load with respect to the local y axis.

$$\begin{aligned} F_q\ =\ & 2.\ \ U^T\ I122(W_\theta^T\ ,\ \dot{W}\ )\ \dot{U}\ \ \ H_{11}\,W_\theta^T g\ \ +\ U^T\ H_{21}\,W_\theta^T g\ - \\ & \mathrm{Trace}(W_\theta\ D_L\ I_L\ D_L^T\ \ddot{W}_r\ +\ 2.W_\theta\ D_L\ I_L\ \dot{D}_L^T\ \dot{W}\ \ +\ A_{R\theta}\,I_R A_{RR}) \\ & +\ \ H_{11}\,D_L^T\ W_\theta^T\ g\ \ +\ T \end{aligned}$$

T is the externally applied torque. A_{RR} is a result of a simplification of the second time derivative of the transformation matrix A_R and is termed D_{L12} residual acceleration. $A_{R\theta}$ is a derivative of A_R with respect to the joint variable theta.

$$\begin{aligned} M_{\eta q}\ =\ & (\ \mathrm{Trace}\ (\ W_\theta\ D_L\ I_L\ D_{L11}^T\ W^T),\ \mathrm{Trace}(W_\theta\ D_L\ I_L\ D_{L12}^T\ W^T)) \\ & \ \ +\ I121(W^T\ ,\ W_\theta) \end{aligned}$$

D_{L11} and D_{L12} are the derivatives of the arm tip deformation transformation matrix differentiated with respect to nodal deflection displacement and the nodal slope displacement respectively.

$$M_{\eta\eta}\ =\ \ I122(W^T\ ,\ W)\ \ +M_L\ \ (\ \ddot{V}(0)\ )\ +\ \ \ (\ I_{xx}+I_{yy})\times(\ \ddot{\Phi}(0)\ )$$

$$K_{\eta} = K_{11} + I122(W^T , \ddot{W}_R)$$

$$\begin{aligned} F_{\eta} = \; & H_{21} W^T g - (\mathrm{Trace}(\ddot{W}_R \, D_L \, I_L \, D_{L11}{}^T W^T + 2.\dot{W} \, \dot{D}_L I_L \, D_{L11}{}^T W^T), \\ & \mathrm{Trace}(\ddot{W}_R \, D_L \, I_L \, D_{L12}{}^T W^T + 2.\dot{W} \, \dot{D}_L I_L \, D_{L12}{}^T W^T)) \; + \\ & H_{41} D_{L11}{}^T W^T g, \; H_{41} D_{L12}{}^T W^T g \end{aligned}$$

These expressions are coded and solved for in the computer program listed in Appendix B.

APPENDIX B

LISTING OF THE DSL CODING UTILIZED IN SOLVING THE DYNAMIC EQUATIONS OF MOTION

```
* THESIS COPY
*
*       SIMULATION OF SINGLE LINK FLEXIBLE MANIPULATOR DYNAMICS
*       -------------------------------------------------------
*
*
*    THIS PROGRAM SOLVES THE ERLS FLEXIBLE MANIPULATOR DYNAMICS FOR A
*    SINGLE LINK EXPERIMENTAL ARM. THE EXPERIMENTAL ARM PARAMETERS ARE
*    INPUTTED AND THE HYDRAULIC ACTUATION DYNAMICS ARE INCLUDED IN THE
*    SIMULATION. THE INPUT IS THE CURRENT TO THE SERVOVALVE MOUNTED ON
*    THE HYDRAULIC ACTUATOR AND THE OUTPUT IS THE POSITION OF THE ARM
*    TIP IN THE GLOBAL REFERENCE SYSTEM. THE CODING CONSISTS OF A MAIN
*    PROGRAM, TEN FUNCTION SUB-PROGRAMS AND EIGHTEEN SUBROUTINES.
*
*    THE FOLLOWING PARAMETERS ARE DEFINED:
*    1.A-EFFECTIVE CROSS-SECTIONAL AREA OF FLEXIBLE ARM
*    2.ARRDD-3X3 SECOND TIME DERIVATIVE OF ROTOR RESIDUAL ACCELERATION
*       MATRIX
*    3.ARTH-3X3 ROTOR TRANSFORMATION MATRIX DIFFERENTIATED WITH RESPECT
*      TO THETA
*    4.A1P-SHAPE FUNCTION MATRIX COEFFICIENT DEFINED IN APPENDIX B
*    5.A2P-SHAPE FUNCTION MATRIX COEFFICIENT DEFINED IN APPENDIX B
*    6.BE-EFFECTIVE BULK MODULUS OF FLUID
*    7.BETA1-SHAPE FUNCTION MATRIX COEFFICIENT DEFINED IN APPENDIX B
*    8.BETA2-SHAPE FUNCTION MATRIX COEFFICIENT DEFINED IN APPENDIX B
*    9.BIGF-3X1 RIGHT-HAND SIDE VECTOR FOR LARGE AND SMALL MOTION
*       ACCELERATIONS
*    10.BIGM-3X3 MATRIX OF LARGE AND SMALL MOTION ACCELERATION
*       COEFFICIENTS
*    11.CTM-TOTAL LEAKAGE COEFFICIENT OF THE ACTUATOR
*    12.C1,C2,C3,C4-SHAPE FUNCTION MATRIX COEFFICIENTS DEFINED IN APPENDIX B
*    16.DEFM-DISPLACEMENT DEFORMATION VARIABLE
*    17.DEFMD-TIME DERIVATIVE OF DISPLACEMENT DEFORMATION VARIABLE
*    18.DIFF,DERR,DERR1,FACTOR-DUMMY VARIABLES
*    19.DL1-3X3 DEFORMATION MATRIX
*    20.DL11-3X3 DEFORMATION MATRIX DIFFERENTIATED WITH RESPECT TO THE
*       DISPLACEMENT DEFORMATION VARIABLE
*    21.DL12-3X3 DEFORMATION MATRIX DIFFERENTIATED WITH RESPECT TO THE
*       SLOPE DEFORMATION VARIABLE
*    22.DL1D-3X3 FIRST TIME DERIVATIVE OF DEFORMATION MATRIX
*    23.DM-ACTUATOR DISPLACEMENT
*    24.E-MODULUS OF ELASTICITY OF STEEL
*    25.FN-2X1 RIGHT-HAND SIDE VECTOR FOR SMALL MOTION ACCELERATIONS
*    26.FQ-RIGHT-HAND SIDE FOR LARGE MOTION ACCELERATIONS
*    27.G-3X1 GRAVITATIONAL ACCELERATION VECTOR
*    28.GPOS-3X1 GLOBAL POSITION VECTOR FOR ARM TIP
*    29.H11-1X3 LINK FIRST MOMENT OF INERTIA VECTOR
*    30.H21-2X3 LINK SHAPE MATRIX FIRST MOMENT OF INERTIAVECTOR
*    31.H41-1X3 LOAD FIRST MOMENT OF INERTIA VECTOR
*    32.KCE-TOTAL FLOW PRESSURE COEFFICIENT
*    33.PL-LOAD HYDRAULIC PRESSURE DROP
*    34.PS-HYDRAULIC SUPPLY PRESSURE
*    35.QL-FLOW DELIVERED FROM THE SERVOVALVE
*    36.SLOP-SLOPE DEFORMATION VARIABLE
*    37.SLOPD-TIME DERIVATIVE OF SLOPE DEFORMATION VARIABLE
*    38.SOL-3X1 VECTOR OF LARGE AND SMALL MOTION ACCELERATIONS
*    39.TE-TORQUE EFFICIENCY
*    40.TH-LARGE MOTION POSITION VARIABLE
*    41.THD-TIME DERIVATIVE OF LARGE MOTION VARIABLE
*    42.TORQUE-APPLIED TORQUE BY ACTUATOR
*    43.U-2X1 ARM TIP DEFORMATION VECTOR INCLUDING DISPLACEMENT AND SLOPE
*    44.UD-2X1 ARM TIP DEFORMATION VECTOR DIFFERENTIATED WITH RESPECT TO
*      TIME
*    45.VT-TOTAL COMPRESSED VOLUME INCLUDING ACTUATOR LINES AND CHAMBERS
*    46.W-3X3 LINK TRANSFORMATION MATRIX
*    47.WD-3X3 FIRST TIME DERIVATIVE OF LINK TRANSFORMATION MATRIX
*    48.WRDD-3X3 SECOND TIME DERIVATIVE OF LINK RESIDUAL ACCELERATION
*       MATRIX
*    49.WTH-3X3 TRANSFORMATION MATRIX DIFFERENTIATED WITH RESPECT TO
*      THETA
*    50.XIFRAC-VARIABLE FRACTIONAL AMOUNT OF INPUT CURRENT TO SERVO-
*       VALVE
*    51.XIINP-CURRENT INPUT EQUAL TO INITIAL AND FRACTIONAL AMOUNTS
*    52.XIL-3X3 INERTIA MATRIX OF THE LOAD
*    53.XIO-INITIAL INPUT CURRENT TO SERVOVALVE
*    54.XIR-3X3 ROTOR INERTIA MATRIX
*    55.XISTEP-STEP INPUT OF FRACTIONAL AMOUNT OF INPUT CURRENT
*    56.XK11-2X2 PARTIAL LINK STIFFNESS MATRIX
*    57.XKN-2X2 LINK STIFFNESS MATRIX
*    58.XKV-SERVOVALVE SIZING CONSTANT
*    59.XLL-LENGTH OF FLEXIBLE ARM
*    60.XML-MASS OF LOAD
*    61.XMNN-2X2 COEFFICIENT MATRIX OF SMALL MOTION ACCELERATIONS IN THE
*       SMALL MOTION DYNAMIC EQUATIONS
*    62.XMNQ-2X1 COEFFICIENT VECTOR OF LARGE MOTION ACCELERATIONS IN THE
*       SMALL MOTION DYNAMIC EQUATIONS
*    63.XMQN-1X2 COEFFICIENT VECTOR OF SMALL MOTION ACCELERATIONS IN THE
*      LARGE MOTION DYNAMICS EQUATION
*    64.XMQQ-COEFFICIENT OF LARGE MOTION ACCELERATION IN THE LARGE MOTION
*      DYNAMICS EQUATION
*    65.XMQQP-2X2 DUMMY MATRIX FOR USE IN FORMULATING THE EQUATIONS OF
*      MOTION
*    66.XMR-MASS OF ACTUATOR ROTOR
*    67.XMU-MASS DENSITY OF STEEL FLEXIBLE ARM
*    68.XMX-FIRST MOMENT OF LOAD WITH RESPECT TO THE LOCAL COORDINATE
*       Y AXIS
*    69.XXI-VARIABLE REPRESENTING INERTIA-LIKE LOAD PROPERTY
*    70.Y-DUMMY VARIABLE TO STORE INTEGRATION RESULTS
*    71.YYI-VARIABLE REPRESENTING INERTIA-LIKE LOAD PROPERTY
*    72.ZI-AREA MOMENT OF INERTIA OF FLEXIBLE ARM
*
INITIAL
```

```
*
*         INITIAL VALUES OF PARAMETERS ARE INPUTTED VIA XINIT SUBROUTINE
*
D       DIMENSION U(2,1),XMQQ(1),XMQQP(2,2),DL1(3,3),WTH(3,3),ARTH(3,3),
D      *XIR(3,3),XMQN(1,2),UD(2,1),H11(1,3),G(3,1),H21(2,3),
D      *WADD(3,3),DL1D(3,3),WD(3,3),ARADD(3,3),H41(1,3),XK11(2,2),
D      *DL12(3,3),XMNQ(2,1),W(3,3),XMNN(2,2),XKN(2,2),FN(2,1),BIGM(3,3),
D      *BIGF(3,1),XIL(3,3),DL11(3,3),DEFMD(1),SOL(3),THD(1),SLOP(1),
D      *SLOPD(1),A(1),E(1),ZI(1),XXI(1),YYI(1),FQ(1),GPOS(3),XITH(1),
D      *XMU(1),XLL(1),XML(1),XMR(1),XMX(1),TH(1),TORQUE(1),DEFM(1),
D      *PS(1),XIFRAC(1),XI0(1),CTM(1),VT(1),BE(1),DM(1),XKV(1),TE(1),
D      *QL(1),PL(1),DIFF(1),XIINP(1),QERR1(1),QERR(1),FACTOR(1),XISTEP(1)

FIXED I
NOSORT
*
*      COMMON BLOCK TO PASS CONSTANTS USED IN THE SHAPE FUNCTION MATRIX
*      DERIVATION OF THE CONSTANTS INCLUDED IN APPENDIX B.
*
D      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
*
       C1=0.515462194
       C2=0.149362234
       C3=-0.107657991
       C4=-0.205906794
       A1P=1.362220557
       A2P=0.981867539
       BETA1=1.877920950
       BETA2=4.701142847
*
*      INITIALIZATION SUBROUTINE
*
       CALL XINIT(TH,THD,DEFM,DEFMD,SLOP,SLOPD,V0,POS0,A,XML,XMU,...
       XLL,XMR,E,ZI,PS,XIFRAC,XI0,CTM,VT,BE,DM,XKV,TE,QL,...
       PL,PLIC)
*
DERIVATIVE
*
*         COEFFICIENTS FOR BOTH LARGE AND SMALL MOTION ACCELERATIONS
*         AND THE RIGHT-HAND SIDES ARE COMPUTED IN THE FOLLOWING
*         SUBROUTINE CALLS. ALSO,THE HYDRAULIC DYNAMICS ARE INCLUDED
*         IN THE MAIN PROGRAM.
*
NOSORT
*
*         HYDRAULIC DYNAMICS  ( MERRITT, CHAPTER SIX )
*         CURRENT TO ACTUATOR IS CHANGED BY CHANGING IFRAC IN XINIT
*
       XISTEP(1)=XIFRAC(1)*STEP(0.0)
       XIINP(1)=XI0(1)+XISTEP(1)
       IF(PL(1).GT.PS(1)) GO TO 2
       GO TO 3
 2     PL(1)=PS(1)
 3     QERR1(1)=(XIINP(1)*XKV(1)*DSQRT(PS(1)-PL(1)))-(DM(1)*THD(1))
       QERR(1)=QERR1(1)/CTM(1)
```

```
      DIFF(1)=QERR(1)-PL(1)
      FACTOR(1)=VT(1)/(4.0D0*BE(1)*CTM(1))
      DIFF1(1)=DIFF(1)
SORT
      PL1=INTGRL(PLIC,DIFF1,1)
NOSORT
      PL(1)=PL1(1)/FACTOR(1)
      TORQUE(1)=TE(1)*PL(1)*DM(1)
*
*     MATRIX AND VECTOR FORMULATION SUBROUTINE
*
      CALL FORM(W,WTH,WD,DL1,DL1D,XIL,XIR,ARTH,WRDD,ARRDD,U,UD,...
      XMQQP,G,H11,H21,DL11,DL12,H41,XK11,A,XMU,XML,XLL,TH,THD,...
      DEFM,DEFMD,SLOP,SLOPD,E,ZI,XMR,XMX,YYI,XXI)
*
*     COEFFICIENT OF LARGE MOTION ACCELERATION IN LARGE MOTION DYNAMICS
*     EQUATION SUBROUTINE
*
      CALL XLMMQQ(XMQQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,A,XMU,TH,DEFM,SLOP)
*
*     COEFFICIENTS OF SMALL MOTION ACCELERATIONS IN LARGE MOTION DYNAMICS
*     EQUATION SUBROUTINE
*
      CALL XLMMQN(XMQN,A,XMU,XML,XLL,XMX,SLOP,DEFM,YYI,XXI)
*
*     RIGHT-HAND SIDE FOR LARGE MOTION DYNAMICS EQUATION SUBROUTINE
*
      CALL XLMFQ(FQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,UD,H11,G,H21,WRDD,...
      DL1D,WD,ARRDD,H41,TH,THD,DEFM,DEFMD,SLOP,SLOPD,A,XMU,XML,XLL,...
      TORQUE)
*
*     LINK STIFFNESS MATRIX SUBROUTINE
*
      CALL SMKN(XKN,XK11,XMQQP,A,XMU,THD)
*
*     COEFFICIENTS OF LARGE MOTION ACCELERATION IN SMALL MOTION
*     DYNAMICS EQUATIONS SUBROUTINE
*
      CALL SMMNQ(XMNQ,DL1,WTH,XIL,DL11,DL12,W,TH,DEFM,SLOP,A,XMU,...
      XLL)
*
*     RIGHT-HAND SIDE OF SMALL MOTION DYNAMICS EQUATIONS SUBROUTINE
*
      CALL SMFN(FN,H21,W,G,WRDD,DL1,XIL,DL11,DL12,WD,DL1D,H41,TH,...
      THD,DEFM,DEFMD,SLOP,SLOPD)
*
*     COEFFICIENTS OF SMALL MOTION ACCELERATIONS IN SMALL MOTION DYNAMICS
*     EQUATIONS SUBROUTINE
*
      CALL SMMNN(XMNN,XMQQP,XML,A,XMU,XXI,YYI,XMX)
*
*     ACCELERATION COEFFICIENTS MATRIX AND RIGHT-HAND SIDE VECTOR
*     FORMULATION SUBROUTINE
*
      CALL BIGFOR(BIGM,BIGF,XMQQ,XMQN,FQ,XMNQ,XMNN,XKN,FN,U)
```

```
*
*     LINEAR EQUATION SOLVER FOR ACCELERATIONS SUBROUTINE
*
      CALL XLEQ(BIGM,BIGF,SOL)
*
*     TRANSFORMATION FROM LOCAL COORDINATE TO GLOBAL COORDINATE TIP
*     POSITION SUBROUTINE
*
      CALL GLOB(GPOS,W,DEFM)
*
*     INTEGRATE ACCELERATIONS AND THEN VELOCITY TO GET LARGE MOTION
*     ANGULAR POSITION AND SMALL MOTION,LOCAL COORDINATE,TIP POSITION
*
      DO 5 I=1,3
      SOL1(I)=SOL(I)
    5 CONTINUE
SORT
      VEL=INTGRL(V0,SOL1,3)
NOSORT
      THD(1)=VEL(1)
      DEFMD(1)=VEL(2)
      SLOPD(1)=VEL(3)
      DO 10 I=1,3
      VEL1(I)=VEL(I)
   10 CONTINUE
SORT
      POS=INTGRL(POS0,VEL1,3)
NOSORT
      TH(1)=POS(1)
      DEFM(1)=POS(2)
      SLOP(1)=POS(3)
      AC1=SOL1(1)
      AC2=SOL1(2)
      AC3=SOL1(3)
      VE1=VEL(1)
      VE2=VEL(2)
      VE3=VEL(3)
      PO1=POS(1)
      PO2=POS(2)
      PO3=POS(3)
      XPOS=GPOS(2)
      YPOS=GPOS(3)
      TORK=TORQUE(1)
      LPD=PL(1)
*
*     OUTPUT GLOBAL COORDINATE TIP POSITION
*
TERMINAL
*     ADAMS METHOD CHOSEN DUE TO FASTER COMPUTATIONAL TIMES,SAME ACCURACY
METHOD ADAMS
PRINT 1.0E-2,XPOS,YPOS
SAVE 1.0E-3,XPOS,YPOS
CONTROL FINTIM=1.00
GRAPH (G1,DE=SPRINT) TIME,XPOS
LABEL (G1) LOAD=2.115 KG
```

```
GRAPH (G2,DE=SPRINT) TIME,YPOS
LABEL (G2) LOAD=2.115 KG
END
STOP
*
*    LISTING OF SUBROUTINES
*
FORTRAN
*
*...........................................................................
*     SUBROUTINE XINIT INTRODUCES INITIAL VALUES AND CONSTANTS INTO THE MAIN
*     BODY OF THE PROGRAM
*
      SUBROUTINE XINIT(TH,THD,DEFM,DEFMD,SLOP,SLOPD,VO,POSO,A,ML,MU,LL,
     #MR,E,ZI,PS,IFRAC,IO,CTM,VT,BE,DM,KV,TE,QL,PL,PLIC)
      REAL*8 VO(3),POSO(3),ML,MU,LL,MR,TH,THD,DEFM,DEFMD,SLOP,SLOPD,
     #A,E,ZI,ITORQ,PS,IFRAC,IO,CTM,VT,BE,DM,KV,TE,QL,PL,PLIC,IMAX,
     #C1,C2,C3,C4,AP1,AP2,BETA1,BETA2

      ITORQ= 44.554909510000000
      DM=6.2271D-05
      TE=.90000000000000
      PL=ITORQ/(DM*TE)
      PLIC=PL
      CTM=3.706477 2D-13
      QL=CTM*PL
      KV=2.402963D-09
      PS=1.37888D+07
      IO=QL/(KV*DSQRT(PS-PL))
      IMAX=10.000000000000000
*
*     CURRENT TO ACTUATOR REGULATED BY FRACTION CHOSEN IN IFRAC
*
      IFRAC=.5D+00*(IMAX-IO)
      VT=3.05127D-04
      BE=690.D6
      A=6.17795D-04
      ML=2.1151000000000
      MU=7861.0500000000000
      LL=0.9985000000000
      MR=9.0001145100000
      E=2.0D11
      ZI=4.065D-10
      VO(1)=0.000000000000000
      VO(2)=0.000000000000000
      VO(3)=0.000000000000000
      POSO(1)=0.00000000000000
      POSO(2)=-.14606414900000
      POSO(3)=-.20078981400000
      TH=POSO(1)
      THD=VO(1)
      DEFM=POSO(2)
      DEFMD=VO(2)
      SLOP=POSO(3)
      SLOPD=VO(3)
      RETURN
      END
*
*.........................................................................................
*     FUNCTION SUBPROGRAM ONE COMPUTES a COEFFICIENT OF SHAPE FUNCTION MATRIX
*
      DOUBLE PRECISION FUNCTION ONE(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      ONE=                               C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                       C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                               (SIN(BETA2*X)+SINH(BETA2*X)))
      RETURN
      END
*
*.............................................................................................
*     FUNCTION SUBPROGRAM TWO COMPUTES  a**2
*
      DOUBLE PRECISION FUNCTION TWO(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      TWO=                              (C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                       C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                               (SIN(BETA2*X)+SINH(BETA2*X))))**2
      RETURN
      END
*
*...............................................................................................
*     FUNCTION SUBPROGRAM THREE COMPUTES b COEFFICIENT OF SHAPE FUNCTION MATRIX
*
      DOUBLE PRECISION FUNCTION THREE(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      THREE=                              C3*(A1P*(COS(BETA1*X)+COSH
     #                     (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C4*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                                (SIN(BETA2*X)+SINH(BETA2*X)))
      RETURN
      END
*
*................................................................................................
*     FUNCTION SUBPROGRAM FOUR COMPUTES b**2
*
      DOUBLE PRECISION FUNCTION FOUR(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      FOUR=                              (C3*(A1P*(COS(BETA1*X)+COSH
     #                     (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C4*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                                (SIN(BETA2*X)+SINH(BETA2*X))))**2
      RETURN
      END
*
*.............................................................................
*     FUNCTION SUBPROGRAM FIVE COMPUTES PRODUCT ab
*

      DOUBLE PRECISION FUNCTION FIVE(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      FIVE=                             (C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                       C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                               (SIN(BETA2*X)+SINH(BETA2*X))))
     #                                  *(C3*(A1P*(COS(BETA1*X)+COSH
     #                     (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C4*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                                (SIN(BETA2*X)+SINH(BETA2*X))))
      RETURN
      END
*
*.............................................................................
*     FUNCTION SUBPROGRAM SIX COMPUTES  -a LL - a x
*

      DOUBLE PRECISION FUNCTION SIX(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      SIX=                        .9985*(C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                       C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                               (SIN(BETA2*X)+SINH(BETA2*X))))+
     #                                X*(C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                       C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                               (SIN(BETA2*X)+SINH(BETA2*X))))
      RETURN
      END
*
*.............................................................................
*     FUNCTION SUBPROGRAM SEVEN COMPUTES  -b LL - b x
*

      DOUBLE PRECISION FUNCTION SEVEN(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      SEVEN=                      .9985* (C3*(A1P*(COS(BETA1*X)+COSH
     #                     (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C4*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                                (SIN(BETA2*X)+SINH(BETA2*X))))
     #                                +X*(C3*(A1P*(COS(BETA1*X)+COSH
     #                     (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C4*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                                (SIN(BETA2*X)+SINH(BETA2*X))))
      RETURN
      END
*
*...........................................................................
*     FUNCTION SUBPROGRAM EIGHT COMPUTES COEFFICIENT OF v" **2
*

      DOUBLE PRECISION FUNCTION EIGHT(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      EIGHT=                            (C1*BETA1*BETA1*(A1P*(-COS
     #         (BETA1*X)+COSH(BETA1*X))+(-SIN(BETA1*X)+SINH(BETA1*X)))+
     #              C2*BETA2*BETA2*(A2P*(-COS(BETA2*X)+COSH(BETA2*X))+
     #                               (-SIN(BETA2*X)+SINH(BETA2*X))))**2
      RETURN
      END
*
*...........................................................................
*     FUNCTION SUBPROGRAM NINE COMPUTES COEFFICIENT OF v"
*

      DOUBLE PRECISION FUNCTION NINE(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      NINE=                             (C1*BETA1*BETA1*(A1P*(-COS
     #         (BETA1*X)+COSH(BETA1*X))+(-SIN(BETA1*X)+SINH(BETA1*X)))+
     #              C2*BETA2*BETA2*(A2P*(-COS(BETA2*X)+COSH(BETA2*X))+
     #                                  (-SIN(BETA2*X)+SINH(BETA2*X))))
     #                                  *(C3*BETA1*BETA1*(A1P*(-COS
     #          (BETA1*X)+COSH(BETA1*X))+(-SIN(BETA1*X)+SINH(BETA1*X)))+
     #               C4*BETA2*BETA2*(A2P*(-COS(BETA2*X)+COSH(BETA2*X))+
     #                                   (-SIN(BETA2*X)+SINH(BETA2*X))))
      RETURN
      END
*
*...........................................................................
*     FUNCTION SUBPROGRAM TEN COMPUTES COEFFICIENT OF v" **2
*

      DOUBLE PRECISION FUNCTION TEN(X)
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,C3,C4,A1P,A2P,BETA1,BETA2

      TEN=                               (C3*BETA1*BETA1*(A1P*(-COS
     #          (BETA1*X)+COSH(BETA1*X))+(-SIN(BETA1*X)+SINH(BETA1*X)))+
     #               C4*BETA2*BETA2*(A2P*(-COS(BETA2*X)+COSH(BETA2*X))+
     #                               (-SIN(BETA2*X)+SINH(BETA2*X))))**2
      RETURN
      END
*
```

```
*......................................................................
*     SUBROUTINE FORM COMPUTES COMPONENTS OF MATRICES
*
      SUBROUTINE FORM(W,WTH,WD,DL1,DL1D,XIL,XIR,ARTH,WRDD,ARRDD,U,UD,
     #XMQOP,G,H11,H21,DL11,DL12,H41,XK11,A,MU,ML,LL,TH,THD,DEFM,DEFMD,
     #SLOP,SLOPD,E,ZI,MR,MX,YYI,XXI)
      REAL*8 W(3,3),WTH(3,3),WD(3,3),DL1(3,3),DL1D(3,3),XIL(3,3)
      REAL*8 XIR(3,3),ARTH(3,3),WRDD(3,3),ARRDD(3,3),U(2,1),UD(2,1)
      REAL*8 XMQQP(2,2),G(3,1),H11(1,3),H21(2,3),DL11(3,3),DL12(3,3)
      REAL*8 H41(1,3),XK11(2,2),MU,ML,LL,MR,MX,TH
      REAL*8 THD,DEFM,DEFMD,SLOP,SLOPD,XXI,YYI,A,E,ZI
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      REAL*S ONE,TWO,THREE,FOUR,FIVE,EIGHT,NINE,TEN,Y
      EXTERNAL ONE,TWO,THREE,FOUR,FIVE,EIGHT,NINE,TEN
      W(1,1)=1.00000000000000
      W(1,2)=0.00000000000000
      W(1,3)=0.00000000000000
      W(2,1)=LL*DCOS(TH)
      W(2,2)=DCOS(TH)
      W(2,3)=-DSIN(TH)
      W(3,1)=LL*DSIN(TH)
      W(3,2)=DSIN(TH)
      W(3,3)=DCOS(TH)
      WTH(1,1)=0.00000000000000
      WTH(1,2)=0.00000000000000
      WTH(1,3)=0.00000000000000
      WTH(2,1)=-LL*DSIN(TH)
      WTH(2,2)=-DSIN(TH)
      WTH(2,3)=-DCOS(TH)
      WTH(3,1)=LL*DCOS(TH)
      WTH(3,2)=DCOS(TH)
      WTH(3,3)=-DSIN(TH)
      WD(1,1)=0.00000000000000
      WD(1,2)=0.00000000000000
      WD(1,3)=0.00000000000000
      WD(2,1)=-LL*DSIN(TH)*THD
      WD(2,2)=-DSIN(TH)*THD
      WD(2,3)=-DCOS(TH)*THD
      WD(3,1)=LL*DCOS(TH)*THD
      WD(3,2)=DCOS(TH)*THD
      WD(3,3)=-DSIN(TH)*THD
      DL1(1,1)=1.00000000000000
      DL1(1,2)=0.00000000000000
      DL1(1,3)=0.00000000000000
      DL1(2,1)=0.00000000000000
      DL1(2,2)=1.00000000000000
      DL1(2,3)=-SLOP
      DL1(3,1)=DEFM
      DL1(3,2)=SLOP
      DL1(3,3)=1.00000000000000
      DL1D(1,1)=0.00000000000000
      DL1D(1,2)=0.00000000000000
      DL1D(1,3)=0.00000000000000
      DL1D(2,1)=0.00000000000000
      DL1D(2,2)=0.0000000000000
```

```
DL1D(2,3)=-SLOPD
DL1D(3,1)=DEFMD
DL1D(3,2)=SLOPD
DL1D(3,3)=0.000000000000000
XIL(1,1)=ML
XIL(1,2)=0.01678545900000
XIL(1,3)=.00000000000000
XIL(2,1)=0.01678545900000
XIL(2,2)=1.776792D-04
XIL(2,3)=0.00000000000000
XIL(3,1)=.00000000000000
XIL(3,2)=0.00000000000000
XIL(3,3)=2.986791D-03
MX=.01678545900000
XXI=1.776790D-04
YYI=2.986791D-03
XIR(1,1)=MR
XIR(1,2)=0.000000000000000
XIR(1,3)=0.000000000000000
XIR(2,1)=0.000000000000000
XIR(2,2)=.027467130000000
XIR(2,3)=0.000000000000000
XIR(3,1)=0.000000000000000
XIR(3,2)=0.000000000000000
XIR(3,3)=.02746713000000
ARTH(1,1)=0.00000000000000
ARTH(1,2)=0.00000000000000
ARTH(1,3)=0.00000000000000
ARTH(2,1)=0.00000000000000
ARTH(2,2)=-DSIN(TH)
ARTH(2,3)=-DCOS(TH)
ARTH(3,1)=0.00000000000000
ARTH(3,2)=DCOS(TH)
ARTH(3,3)=-DSIN(TH)
WRDD(1,1)=0.00000000000000
WRDD(1,2)=0.00000000000000
WRDD(1,3)=0.00000000000000
WRDD(2,1)=-LL*DCOS(TH)*(THD**2)
WRDD(2,2)=-DCOS(TH)*(THD**2)
WRDD(2,3)=DSIN(TH)*(THD**2)
WRDD(3,1)=-LL*DSIN(TH)*(THD**2)
WRDD(3,2)=-DSIN(TH)*(THD**2)
WRDD(3,3)=-DCOS(TH)*(THD**2)
ARRDD(1,1)=0.00000000000000
ARRDD(1,2)=0.00000000000000
ARRDD(1,3)=0.00000000000000
ARRDD(2,1)=0.00000000000000
ARRDD(2,2)=-DCOS(TH)*(THD**2)
ARRDD(2,3)=DSIN(TH)*(THD**2)
ARRDD(3,1)=0.00000000000000
ARRDD(3,2)=-DSIN(TH)*(THD**2)
ARRDD(3,3)=-DCOS(TH)*(THD**2)
U(1,1)=DEFM
U(2,1)=SLOP
UD(1,1)=DEFMD
```

```
      UD(2,1)=SLOPD
      CALL DQG4(-LL,0.D0,TWO,Y)
      XMQQP(1,1)=Y
      CALL DQG4(-LL,0.D0,FIVE,Y)
      XMQQP(1,2)=Y
      XMQOP(2,1)=Y
      CALL DQG4(-LL,0.D0,FOUR,Y)
      XMQQP(2,2)=Y
      G(1,1)=0.00000000000000
      G(2,1)=0.00000000000000
      G(3,1)=-9.80660000000000
      H11(1,1)=4.85651900000000
      H11(1,2)=-2.42825869300000
      H11(1,3)=0.00000000000000
      H21(1,1)=0.00000000000000
      H21(1,2)=0.00000000000000
      CALL DQG4(-LL,-0.D0,ONE ,Y)
      H21(1,3)=A*MU*Y
      H21(2,1)=0.00000000000000
      H21(2,2)=0.00000000000000
      CALL DQG4(-LL,0.D0,THREE,Y)
      H21(2,3)=A*MU*Y
      DO 50 I=1,3
      DO 60 J=1,3
      DL11(I,J)=0.00000000000000
      DL12(I,J)=0.00000000000000
   60 CONTINUE
   50 CONTINUE
      DL11(3,1)=1.00000000000000
      DL12(2,3)=-1.00000000000000
      DL12(3,2)=1.00000000000000
      H41(1,1)=ML
      H41(1,2)=0.01678545900000
      H41(1,3)=.000000000000000000
      CALL DQG4(-LL,0.D0,EIGHT,Y)
      XK11(1,1)=Y*E*ZI
      CALL DQG4(-LL,0.D0,NINE,Y)
      XK11(1,2)=Y*E*ZI
      XK11(2,1)=Y*E*ZI
      CALL DQG4(-LL,0.D0,TEN ,Y)
      XK11(2,2)=Y*E*ZI
      RETURN
      END
*
*...........................................................................
*     SUBROUTINE XLMMQQ COMPUTES THE COEFFICIENTS OF LARGE MOTION ACCELERATION
*     IN THE LARGE MOTION DYNAMICS EQUATION
*
      SUBROUTINE XLMMQQ(MQQ,U,XMQOP,DL1,WTH,ARTH,XIL,XIR,A,MU,TH,DEFM,
     #SLOP)
      REAL*8 MQQ,UT(1,2),P(1,2),DL1T(3,3),WTHT(3,3),ARTHT(3,3),P1(3,3)
      REAL*8 P2(3,3),P3(3,3),P4(3,3),P5(3,3),P6(3,3),P7(3,3),MU
      REAL*8 U(2,1),XMQQP(2,2),DL1(3,3),WTH(3,3),ARTH(3,3),XIL(3,3)
      REAL*8 XIR(3,3),A,TH,DEFM,SLOP,SP,TP
      M=2
      L=1
      N=3
      MQQ=0.00000000000000
      CALL TRANS(U,UT,M,L)
      CALL MATMUL(UT,XMQQP,L,M,M,P)
      CALL MATMUL(P,U,L,M,L,SP)
      CALL TRANS(DL1,DL1T,N,N)
      CALL TRANS(ARTH,ARTHT,N,N)
      CALL TRANS(WTH,WTHT,N,N)
      CALL MATMUL(WTH,DL1,N,N,N,P1)
      CALL MATMUL(P1,XIL,N,N,N,P2)
      CALL MATMUL(P2,DL1T,N,N,N,P3)
      CALL MATMUL(P3,WTHT,N,N,N,P4)
      CALL MATMUL(ARTH,XIR,N,N,N,P5)
      CALL MATMUL(P5,ARTHT,N,N,N,P6)
      CALL MATADD(P4,P6,N,N,P7)
      CALL TRACE(P7,N,TP)
      MQQ=((1./3.)*A*MU) + (A*MU*SP) + TP
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE XLMMQN COMPUTES THE SMALL MOTION ACCELERATIONS IN THE LARGE
*     IN THE LARGE MOTION DYNAMICS EQUATION
*
      SUBROUTINE XLMMQN(XMQN,A,MU,ML,LL,MX,SLOP,DEFM,YYI,XXI)
      REAL*8 XMQN(1,2),MU,ML,LL,MX,A,SLOP,DEFM,YYI,XXI
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2,Y,SIX,SEVEN
      EXTERNAL SIX,SEVEN
      CALL DQG4(-LL,0.D0,SIX ,Y)
      XMQN(1,1)=(A*MU*Y)+(ML*LL)+MX
      CALL DQG4(-LL,0.D0,SEVEN,Y)
      XMQN(1,2)=(A*MU*Y)+(MX*LL)+YYI+XXI
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE XLMFQ FORMULATES THE RIGHT HAND SIDE FOR THE LARGE MOTION
*     DYNAMICS EQUATION
*
      SUBROUTINE XLMFQ(FQ,U,XMQQF,DL1,WTH,ARTH,XIL,XIR,UD,H11,G,H21,
     #WRDD,DL1D,WD,ARRDD,H41,TH,THD,DEFM,DEFMD,SLOP,SLOPD,A,MU,ML,LL,
     #TORQUE)
      REAL*8 FQP(2,2),P(1,2),P1(1,3),P2(1,3),P3(1,3),P4(3,3),P5(3,3)
      REAL*8 P6(3,3),P7(3,3),P8(3,3),P9(3,3),P10(3,3),P11(1,3),P12(1,3)
      REAL*8 FPFH(3,3),FHP(3,3),FPT(3,3),DL1DT(3,3),WDT(3,3),ARRDDT(3,3)
      REAL*8 UT(1,2),DL1T(3,3),WRDDT(3,3),FPF(3,3),FPS(3,3),WTHT(3,3)
      REAL*8 U(2,1),XMQQP(2,2),DL1(3,3),WTH(3,3),ARTH(3,3),XIL(3,3)
      REAL*8 XIR(3,3),UD(2,1),H11(1,3),G(3,1),H21(2,3),WRDD(3,3)
      REAL*8 DL1D(3,3),WD(3,3),ARRDD(3,3),H41(1,3),MU,LL,ML
      REAL*8 A,TORQUE,FQ,TH,THD,DEFM,DEFMD,SLOP,SLOPD
      REAL*8 FP,SP,TP,TFP,FTHP
      M=2
      L=1
      N=3
      CALL TRANS(U,UT,M,L)
      DO 10 I=1,2
      DO 20 J=1,2
      FQP(I,J)=XMQQP(I,J)*THD
   20 CONTINUE
   10 CONTINUE
      CALL MATMUL(UT,FQP,L,M,M,P)
      CALL MATMUL(P,UD,L,M,L,FP)
      CALL TRANS(WTH,WTHT,N,N)
      CALL MATMUL(H11,WTHT,L,N,N,P1)
      CALL MATMUL(P1,G,L,N,L,SP)
      CALL MATMUL(UT,H21,L,M,N,P2)
      CALL MATMUL(P2,WTHT,L,N,N,P3)
      CALL MATMUL(P3,G,L,N,L,TP)
      CALL TRANS(DL1,DL1T,N,N)
      CALL TRANS(WRDD,WRDDT,N,N)
      CALL MATMUL(WTH,DL1,N,N,N,P4)
      CALL MATMUL(P4,XIL,N,N,N,P5)
      CALL MATMUL(P5,DL1T,N,N,N,P6)
      CALL MATMUL(P6,WRDDT,N,N,N,FPF)
      CALL TRANS(DL1D,DL1DT,N,N)
      CALL TRANS(WD,WDT,N,N)
      CALL MATMUL(WTH,DL1,N,N,N,P7)
      CALL MATMUL(P7,XIL,N,N,N,P8)
      CALL MATMUL(P8,DL1DT,N,N,N,P9)
      CALL MATMUL(P9,WDT,N,N,N,FPS)
      CALL TRANS(ARRDD,ARRDDT,N,N)
      CALL MATMUL(ARTH,XIR,N,N,N,P10)
      CALL MATMUL(P10,ARRDDT,N,N,N,FPT)
      DO 30 I=1,3
      DO 40 J=1,3
      FPS(I,J)=FPS(I,J)*2.
   40 CONTINUE
   30 CONTINUE
      CALL MATADD(FPF,FPS,N,N,FPFH)
      CALL MATADD(FPFH,FPT,N,N,FHP)
      CALL TRACE(FHP,N,TFP)
      CALL MATMUL(H41,DL1T,L,N,N,P11)
      CALL MATMUL(P11,WTHT,L,N,N,P12)
      CALL MATMUL(P12,G,L,N,L,FTHP)
      FQ=(-2.*A*MU*FP) + SP + TP - TFP + FTHP + TORQUE
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE SMKN CALCULATES THE LINK STIFFNESS MATRIX
*
      SUBROUTINE SMKN(XKN,XK11,XMQQP,A,MU,THD)
      REAL*8 XKN(2,2),KNP(2,2),XMQQP(2,2),XK11(2,2),A,THD,MU
      DO 10 I=1,2
      DO 20 J=1,2
      KNP(I,J)=XMQQP(I,J)*(-A)*MU*(THD**2)
      XKN(I,J)=KNP(I,J)+XK11(I,J)
   20 CONTINUE
   10 CONTINUE
```

```
      RETURN
      END
*
*.........................................................................
*     SUBROUTINE SMMNQ COMPUTES THE COEFFICIENTS OF LARGE MOTION ACCELERATION
*     IN THE SMALL MOTION DYNAMICS EQUATION
*
      SUBROUTINE SMMNQ(XMNQ,DL1,WTH,XIL,DL11,DL12,W,TH,DEFM,SLOP,A,MU,
     #LL)
      REAL*8 XMNQ(2,1),DL12T(3,3),DL11T(3,3),WT(3,3),P1(3,3),P2(3,3)
      REAL*8 P3(3,3),P4(3,3),P5(3,3),P6(3,3),DL1(3,3),WTH(3,3),XIL(3,3)
      REAL*8 W(3,3),DL11(3,3),DL12(3,3),TH,DEFM,SLOP,A,MU,LL
      REAL*8 C1,C2,C3,C4,A1P,A2P,BETA1,BETA2
      REAL*8 SIX,SEVEN,Y
      EXTERNAL SIX,SEVEN
      M=2
      L=1
      N=3
      CALL TRANS(DL11,DL11T,N,N)
      CALL TRANS(DL12,DL12T,N,N)
      CALL TRANS(W,WT,N,N)
      CALL MATMUL(WTH,DL1,N,N,N,P1)
      CALL MATMUL(P1,XIL,N,N,N,P2)
      CALL MATMUL(P2,DL11T,N,N,N,P3)
      CALL MATMUL(P3,WT,N,N,N,P4)
      CALL TRACE(P4,N,TFP1)
      CALL DQG4(-LL,0.D0,SIX ,Y)
      XMNQ(1,1)=TFP1 + (Y*A*MU)
      CALL MATMUL(P2,DL12T,N,N,N,P5)
      CALL MATMUL(P5,WT,N,N,N,P6)
      CALL TRACE(P6,N,TFP2)
      CALL DQG4(-LL,0.D0,SEVEN,Y)
      XMNQ(2,1)=TFP2 + (Y*A*MU)
      RETURN
      END
*
*.........................................................................
*     SUBROUTINE SMFN CALCULATES THE RIGHT HAND SIDE OF THE SMALL MOTION
*     DYNAMICS EQUATION
*
      SUBROUTINE SMFN(FN,H21,W,G,WADD,DL1,XIL,DL11,DL12,WD,DL1D,H41,TH,
     #THD,DEFM,DEFMD,SLOP,SLOPD)
      REAL*8 FN(2,1),P1(2,3),P2(3,3),P3(3,3),P4(3,3),P5(3,3),P6(3,3)
      REAL*8 P7(3,3),P8(3,3),P9(3,3),P10(3,3),P11(3,3),P12(3,3),P13(3,3)
      REAL*8 P14(1,3),P15(1,3),P16(1,3),P17(1,3),TP(2,1),FP(2,1),SP(2,1)
      REAL*8 FN1(3,3),FN2(3,3),G(3,1),H21(2,3),WADD(3,3),DL1D(3,3)
      REAL*8 WD(3,3),H41(1,3),XIL(3,3),W(3,3),DL11(3,3),DL12(3,3)
      REAL*8 DL1(3,3),DL11T(3,3),DL12T(3,3),WT(3,3)
      REAL*8 TH,THD,DEFM,DEFMD,SLOP,SLOPD
      M=2
      L=1
      N=3
      CALL TRANS(W,WT,N,N)
      CALL MATMUL(H21,WT,M,N,N,P1)
      CALL MATMUL(P1,G,M,N,L,FP)
      CALL TRANS(DL11,DL11T,N,N)
      CALL TRANS(DL12,DL12T,N,N)
      CALL MATMUL(WRDD,DL1,N,N,N,P2)
      CALL MATMUL(P2,XIL,N,N,N,P3)
      CALL MATMUL(P3,DL11T,N,N,N,P4)
      CALL MATMUL(P4,WT,N,N,N,P5)
      CALL MATMUL(WD,DL1D,N,N,N,P6)
      CALL MATMUL(P6,XIL,N,N,N,P7)
      CALL MATMUL(P7,DL11T,N,N,N,P8)
      CALL MATMUL(P8,WT,N,N,N,P9)
      CALL MATMUL(P3,DL12T,N,N,N,P10)
      CALL MATMUL(P10,WT,N,N,N,P11)
      CALL MATMUL(P7,DL12T,N,N,N,P12)
      CALL MATMUL(P12,WT,N,N,N,P13)
      DO 10 I=1,3
      DO 20 J=1,3
      P9(I,J)= P9(I,J)*2.
      P13(I,J)=P13(I,J)*2.
  20  CONTINUE
  10  CONTINUE
      CALL MATADD(P5,P9,N,N,FN1)
      CALL MATADD(P11,P13,N,N,FN2)
      CALL TRACE(FN1,N,TFN1)
      CALL TRACE(FN2,N,TFN2)
      SP(1,1)=TFN1
      SP(2,1)=TFN2
      CALL MATMUL(H41,DL11T,L,N,N,P14)
      CALL MATMUL(P14,WT,L,N,N,P15)
      CALL MATMUL(P15,G,L,N,L,FN3)
      CALL MATMUL(H41,DL12T,L,N,N,P16)
      CALL MATMUL(P16,WT,L,N,N,P17)
      CALL MATMUL(P17,G,L,N,L,FN4)
      TP(1,1)=FN3
      TP(2,1)=FN4
      DO 31 I=1,2
      FN(I,1)=FP(I,1) - SP(I,1) + TP(I,1)
 31   CONTINUE
      RETURN
      END
*
*....................................................................................
*     SUBROUTINE SMMNN COMPUTES SMALL MOTION ACCELERATIONS IN THE SMALL MOTION
*     DYNAMICS EQUATION
*
      SUBROUTINE SMMNN(XMNN,XMQQP,ML,A,MU,XXI,YYI,MX)
      REAL*8 XMNN(2,2),XMQQP(2,2),ML,MU,A,XXI,YYI,MX
      DO 10 I=1,2
      DO 20 J=1,2
      XMNN(I,J)=0.0000000000000
  20  CONTINUE
  10  CONTINUE
      XMNN(1,1)=ML
      XMNN(1,2)=MX
      XMNN(2,1)=MX
      XMNN(2,2)=XXI+YYI
      DO 30 I=1,2
      DO 40 J=1,2
      XMNN(I,J)=XMNN(I,J) + XMQQP(I,J)*A*MU
   40 CONTINUE
  30  CONTINUE
      RETURN
      END
*
*.......................................................................
*     MULTIPLICATION SUBROUTINE
*
      SUBROUTINE MATMUL(A,B,M,L,N,C)
      REAL*8 A(M,L),B(L,N),C(M,N)
      DO 10 I=1,M
      DO 20 J=1,N
      C(I,J)=0.0
      DO 30 INDEX=1,L
      C(I,J)=C(I,J) + A(I,INDEX)*B(INDEX,J)
  30  CONTINUE
  20  CONTINUE
 10   CONTINUE
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE TRANS PERFORMS TRANSPOSE OPERATION
*
      SUBROUTINE TRANS(A,B,M,L)
      REAL*8 A(M,L),B(L,M)
      DO 10 I=1,M
      DO 20 J=1,L
      B(J,I)=A(I,J)
 20   CONTINUE
 10   CONTINUE
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE TRACE CALCULATES THE TRACE OF A MATRIX
*
      SUBROUTINE TRACE(A,M,TRAC)
      REAL*8 A(M,M)
      TRAC=0.0
      DO 10 I=1,M
      TRAC=TRAC + A(I,I)
  10  CONTINUE
      RETURN
      END
*
*.......................................................................
*     MATRIX ADDITION SUBROUTINE
*
      SUBROUTINE MATADD(A,B,M,L,C)
      REAL*8 A(M,L),B(M,L),C(M,L)
      DO 10 I=1,M
      DO 20 J=1,L
      C(I,J)=A(I,J) + B(I,J)
 20   CONTINUE
 10   CONTINUE
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE BIGFOR
*
      SUBROUTINE BIGFOR(BIGM,BIGF,MQQ,XMQN,FQ,XMNQ,XMNN,XKN,FN,U)
      REAL*8 BIGM(3,3),BIGF(3,1),XMQN(1,2),XMNQ(2,1),XMNN(2,2),XKN(2,2)
      REAL*8 FN(2,1),U(2,1),MQQ,P(2,1),FQ
      M=2
      L=1
      BIGM(1,1)=MQQ
      BIGM(1,2)=XMQN(1,1)
      BIGM(1,3)=XMQN(1,2)
      BIGM(2,1)=XMNQ(1,1)
      BIGM(2,2)=XMNN(1,1)
      BIGM(2,3)=XMNN(1,2)
      BIGM(3,1)=XMNQ(2,1)
      BIGM(3,2)=XMNN(2,1)
      BIGM(3,3)=XMNN(2,2)
      BIGF(1,1)=FQ
      CALL MATMUL(XKN,U,M,M,L,P)
      BIGF(2,1)=FN(1,1)-P(1,1)
      BIGF(3,1)=FN(2,1)-P(2,1)
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE XLEQ CALL IMSL SUBROUTINE LEQT2F TO SOLVE SYSTEM OF EQUATIONS
*
      SUBROUTINE XLEQ(BIGM,BIGF,SOL)
      REAL*8 BIGM(3,3),BIGF(3,1),SOL(3),WKAREA(18)
      M=1
      N=3
      CALL LEQT2F (BIGM,M,N,N,BIGF,M,WKAREA,IER)
      DO 10 I=1,3
      SOL(I)=BIGF(I,1)
  10  CONTINUE
      RETURN
      END
*
*.......................................................................
*     SUBROUTINE GLOB PERFORMS TRANSFORMATION FROM RELATIVE POSITION TO
*     GLOBAL POSITION
*
      SUBROUTINE GLOB(GPOS,W,DEFM)
      REAL*8 GPOS(3),W(3,3),DEFM,RL(3)
      RL(1)=1.0D0
      RL(2)=0.0D0
      RL(3)=DEFM
      N=3
      L=1
      CALL MATMUL(W,AL,N,N,L,GPOS)
      RETURN
      END

*     ....................................................................
*
*        SUBROUTINE DQG4 (IMSL SUBROUTINE)
*
*        PURPOSE
*           TO COMPUTE INTEGRAL(FCT(X), SUMMED OVER X FROM XL TO XU)
*
*        USAGE
*           CALL DQG4 (XL,XU,FCT,Y)
*           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
*        DESCRIPTION OF PARAMETERS
*           XL     - DOUBLE PRECISION LOWER BOUND OF THE INTERVAL.
*           XU     - DOUBLE PRECISION UPPER BOUND OF THE INTERVAL.
*           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
*                    SUBPROGRAM USED.
*           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
*        METHOD
*           EVALUATION IS DONE BY MEANS OF 4-POINT GAUSS QUADRATURE
*           FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 7
*           EXACTLY. FOR REFERENCE, SEE
*           V.I.KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS,
*           MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340.
*
      SUBROUTINE DQG4(XL,XU,FCT,Y)
*
      DOUBLE PRECISION XL,XU,Y,A,B,C,FCT
      A=.5D0*(XU+XL)
      B=XU-XL
      C=.43056815579702629D0*B
      Y=.17392742256872693D0*(FCT(A+C)+FCT(A-C))
      C=.16999052179242813D0*B
      Y=B*(Y+.32607257743127307D0*(FCT(A+C)+FCT(A-C)))
      RETURN
      END
```

APPENDIX C

DATA REDUCTION ALGORITHM

The accelerometer senses an global acceleration, we will call it S, since motion in the vertical plane involves both X and Y components of acceleration. In order to obtain tip position information the global transformation matrix can be used to to obtain an expression for $\ddot{V}(0)$. This is then integrated twice using a DSL simulation program. The value of $v(0)$ is then placed in the global transformation matrix to obtain an expression for tip position in terms of the global coordinate system. The process is repeated below.

Using the global transformation matrix we obtain a value for x and y in terms of v.

$$x = L \cos \theta - v \sin \theta \qquad \text{(C-1)}$$

$$y = v \cos \theta + L \sin \theta \qquad \text{(C-2)}$$

these expressions are then differentiated,

$$\dot{x} = -L \sin \theta \, \dot{\theta} - \dot{v} \sin \theta - v \cos \theta \, \dot{\theta} \qquad \text{(C-3)}$$

$$\dot{y} = -v \sin \theta \, \dot{\theta} + \dot{v} \cos \theta + L \, \dot{\theta} \cos \theta \qquad \text{(C-4)}$$

these expressions are differentiated once again,

$$\ddot{x} = -L\dot{\theta}^2 - L\ddot{\theta}\sin\theta + v\dot{\theta}^2\sin\theta - v\ddot{\theta}\cos\theta - \dot{v}\dot{\theta}\cos\theta$$
$$- \dot{v}\dot{\theta}\cos\theta - \ddot{v}\sin\theta \qquad \text{(C-5)}$$

$$\ddot{y} = -L\dot{\theta}^2\sin\theta + L\ddot{\theta}\cos\theta - v\dot{\theta}^2\cos\theta - v\ddot{\theta}\sin\theta$$
$$- \dot{v}\dot{\theta}\sin\theta - \dot{v}\dot{\theta}\sin\theta + \ddot{v}\cos\theta \qquad \text{(C-6)}$$

This expression can be simplified since for our model $\ddot{\theta} = 0$, $\dot{\theta} =$ constant, $L =$ constant.

(C-7)

$$\ddot{x} = \ddot{v}(-\sin\theta) + \dot{v}(-2\dot{\theta}\cos\theta) + v(\dot{\theta}^2\sin\theta) - (L\dot{\theta}^2\cos\theta)$$
$$\ddot{y} = \ddot{v}(\cos\theta) + \dot{v}(-2\dot{\theta}\sin\theta) + v(-\dot{\theta}^2\cos\theta) - (L\dot{\theta}^2\sin\theta)$$

(C-8)

now express in terms of $\ddot{v}$,

(C-9)

$$\ddot{v} - 2\dot{\theta}^2\ddot{v}v + 4\dot{\theta}^2\dot{v}\dot{v} + \dot{\theta}^4 v v + 4L\dot{\theta}^3\dot{v} + L^2\dot{\theta}^4 - S^2 = 0$$

where

$$S^2 = \ddot{x}^2 + \ddot{y}^2$$

this expression is coded in a DSL program and (C-9) is solved explicitly for v and integrated twice. This value of v is then placed in expressions (C-1) and (C-2) and global tip position is determined.

LIST OF REFERENCES

1. Everett, H. R., "Robotics in the Navy Part II: Nonindustrial Development Efforts", Robotics Age The Journal of Intelligent Machines, v.7, pp. 11-14, December 1985.

2. Roth, B., "Robots-State of the Art in Regard to Mechanism Theory," Journal of Mechanism, Transmissions, and Automation in Design, v.105, pp. 11-12, March 1983.

3. Schmitz, E., Experiments on the End-Point Position Control of a Very Flexible One-Link Manipulator, Ph.D. Thesis, Stanford University Dept. of Aeronautics and Astronautics, June 1985.

4. Usoro, P.B., and Nadira, R., "Analysis of Lightweight Flexible Manipulator Dynamics", ASME Proceedings of International Computers in Engineering Conference and Exhibit, pp. 167-174, 1984.

5. Chang, L., Dynamic Analysis of Robotic Manipulators with Flexible Links, Ph.D. Thesis, Purdue University, December 1984.

6. Book, W.J., "Analysis of Massless Elastic Chains with Servo Controlled Joints", ASME Journal of Dynamic Systems, Measurement, and Control, v.101, pp. 187-192, September 1979

7. Sunada, W.H., and Dubowsky, S., "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members", ASME Journal of Mechanisms, Transmisssions, and Automation in Design, v.105, pp. 42-51, March 1983.

8. Book, W. J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", The International Journal of Robotic Research, v.3, Fall, 1984.

9. Hastings, G., and Book, W., "Experiments in the Control of a Flexible Robot Arm", Proceedings, ROBOT9 Conference, v.2, RI/SME, pp. 20-45--20-47, June 1985.

10. Truckenbrodt, A., "Modelling and Control of Flexible Manipulator Structures", Proceedings of 4TH CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators, Warsaw, pp. 110-120, 1981.

11. Cannon, R., and Schmitz, E., "Initial Experiments on the End-Point Control of a Flexible One-Link Robot", The International Journal of Robotic Research, v.3, Fall, 1984.

12. Petroka, R. P., Computer Simulation And Experimental Validation Of A Dynamic Model (Equivalent Rigid Link System) On A Single-Link Flexible Manipulator, M.S. Thesis, Naval Postgraduate School, Monterey, California, October 1982.

13. MOOG Series 760 Two-Stage Flow Control Servovalve Catalog, MOOG INC., East Aurora, New York.

14. Merritt, H.E., Hydraulic Controls Systems, Wiley, New York, 1967.

15. MacADIOS Model 411 Hardware Unit Users Manual, GWI Instruments, Cambridge, Massachusets

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5002

3. Department Chairman, Code 69Hy 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943

4. Professor Liang-Wey Chang, Code 69Ck 2
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943

5. Naval Surface Weapons Center, White Oak 1
Code R402 (ATTN: Mary Lacey)
New Hampshire Avenue
Silver Spring, Maryland 20910

6. LT Kevin P.Gannon 5
2000 Swan Terrace
Alexandria, Virginia 22307

221012

Thesis
G14395 Gannon
c.1 Modeling and experimental validation of a single-link flexible manipulator.